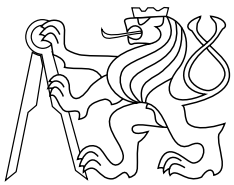
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD Thesis Proposal

Scene text recognition in images and video

Lukáš Neumann

neumalu1@cmp.felk.cvut.cz

August 31, 2012

Thesis Advisor: prof. Jiří Matas

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Abstract

Methods for scene text localization and recognition aim to find all areas in an image (or a video) that would be considered as text by a human, mark boundaries of the areas (usually by rectangular bounding boxes) and output a sequence of (Unicode) characters associated with its content. They allow for real-world images and video processing (i.e. processing of images/videos taken by a standard camera or a mobile phone) and "reading" content of each detected area into a digital text format, that can be further processed by a computer.

We have proposed an end-to-end real-time scene text localization and recognition method which achieves state-of-the-art results on standard datasets (we consider a text recognition method real-time if the processing time is comparable with the time it would take a human to read the text).

Finally, our goals and future work is outlined and it is shown that there is still considerable space for further research and improvements as the overall recall on the standard dataset is only 65%.

Contents

1 Introduction **3**

2 Previous Work **5**
2.1 General text localization 5
2.1.1 Methods based on connected components analysis . . . 5
2.1.2 Sliding window methods 7
2.2 Lexicon-based word spotting 8
2.3 Cropped text recognition 9
2.4 Correcting text recognition 10

3 Our Work **10**

4 Goals of the Thesis **11**

A A method for text localization and recognition in real-world images **12**

B Estimating hidden parameters for text localization and recognition **27**

C Text Localization in Real-world Images using Efficiently Pruned Exhaustive Search **36**

D Real-Time Scene Text Localization and Recognition **42**

1 Introduction

Methods for scene text localization and recognition aim to find all areas in an image (or a video) that would be considered as text by a human, mark boundaries of the areas (usually by rectangular bounding boxes) and output a sequence of (Unicode) characters associated with its content. They allow for real-world images and video processing (i.e. processing of images/videos taken by a standard camera or a mobile phone) and "reading" content of each detected area into a digital text format that can be further processed by a computer.

Scene text localization and recognition (also known as *text localization and recognition in real-world images*, *nature scene OCR* or *text-in-the-wild* problem) is an open problem, unlike printed document recognition where state-of-the-art systems are able to recognize correctly more than 99% of characters [20] (see Figure 1). 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.

Applications of text localization and recognition in real-world images range from automatic annotation of image databases based on their textual content (e.g. Flickr or Google Images), assisting the visually impaired to reading labels on businesses in map applications (e.g. Google Street View).

(a) (b)

Figure 1: The difference between printed document recognition (OCR) and scene text localization and recognition. (a) a scanned book page. (b) a sample from the ICDAR 2003 Robust Reading dataset [22]

Existing methods for general text localization can be categorized into two major groups - methods based on a sliding window and methods based on regions (characters) grouping. Methods in the first category [4, 18, 15] use a window which is moved over the image and the presence of text is estimated on the basis of local image features. While these methods are generally more robust to noise in the image, their computational complexity is high because of the need to search with many rectangles of different sizes, aspect ratios and potentially rotations. Additionally, support for slanted or perspectively distorted text is limited and sliding window methods do not always provide accurate enough text segmentation which can be used for character recognition [4].

The majority of recently published methods for text localization falls into the latter category [35, 10, 26, 47, 45, 25]. The methods differ in their approach to individual character detection, which could be based on edge detection, character energy calculation or extremal region detection. While the methods are paying great attention to individual character detection, grouping of individual characters into words is performed based on heuristics or graph optimization methods and only unary and pairwise constraints are used.

All the methods listed above are focused solely on text localization, i.e. they estimate position of the text, but do not provide its content. Our method (first presented in [26]) was the first one to show end-to-end text localization and recognition and only a few other methods that perform both text localization and recognition have been published since. The method of Wang et al. [42] finds individual characters as visual words using the sliding-window approach and then uses a lexicon to group characters into words. The method is able to cope with noisy data, but its generality is limited as a lexicon of words (which contains at most 500 words in their experiments) has to be supplied for each individual image.

Several competitions [22, 21, 39] have been held in this field to evaluate text localization performance of the methods. Sadly, end-to-end text localization and recognition was not part even of the most recent competition ([39]), despite our requests to the organizers.

The rest of the document is structured as follows. In the Section 2 previous work and the state of the art is presented, in the Section 3 our work is presented and finally in the Section 4.

2 Previous Work

2.1 General text localization

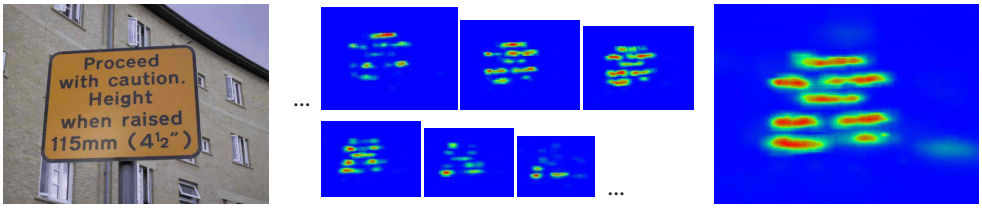
2.1.1 Methods based on connected components analysis

Methods in this category apply a bottom-up strategy for text localization. At first, certain local features are calculated for each pixel in the image and then pixels with similar feature values are grouped together using connected component analysis to form characters, assuming low variance of the used feature(s) within a single character. The methods can be scale-invariant and they inherently provide character segmentation, which can be then used in an OCR stage. The biggest drawback is a sensitivity to noisy and low-resolution images, because they require low variance of local features in sufficient number of pixels.

One of the first methods for general character localization was introduced by Ohya et al. [32]. In their method, a local adaptive thresholding in grey-scale images is used to detect candidate regions and regions with sufficient contrast are selected as characters. Li et al. [16] apply thresholding in a quantized color space and they group individual characters into text blocks by simple alignment rules. Both methods assume that characters are upright without any rotation, that the contrast is high and that background is uniform, which may be sufficient for signs or licence plates, but not for general text localization.

Kim et al. [13] combine three independent detection channels (color continuity, edge detection and color variance) to find candidate regions. Candidate regions are grouped into blocks by size and position constraints and each block is then divided into overlapping 16×16 pixel subblocks, which are verified by a trained SVM classifier [7] using wavelet transform to generate features. If a ratio of subblocks marked as text is higher than a predefined threshold, the block is marked as text. The method is not scale-invariant because of the constant size of subblocks used for classification and the precision is relatively bad because many small text-like patches are detected.

Takahashi and Nakajima [41] use a graph scheme, where vertices represent characters and edges represent a predecessor-successor relation in a block of text. Candidate regions are found using Canny edge detector in the CIELUV color space [5] and regions that pass a set of heuristic constraints are considered vertices of the graph. Edges are created between neighboring vertices and each edge is assigned a weight based on spatial distance, shape and area similarity. Finally, a minimum spanning tree (MST) algorithm is applied and edges with distance or angle over a predefined threshold are re-



(a) (b) (c)

Figure 2: Text confidence map. (a) the original image. (b) text confidence maps for the image pyramid. (c) the text confidence map for the original image. *Images adopted from [35]*

moved. The method cannot handle illumination variations in foreground or background and optimal edge weight estimation remains an open question.

Pan et al. [35, 36] create a text confidence map (see Figure 2) on a grey-scale image pyramid using a calibrated Waldboost [40] classifier with Histogram Of Gradient (HOG) features. Candidate regions are detected independently on a grey-scale image using Niblack's binarization algorithm [31] and a Conditional Random Field (CRF) [14] is employed to label regions as text or non-text, considering the text confidence as one of the unary features. Then, a simple gradient graph energy minimization approach is applied to form block of texts. The method is computationally expensive (because of the image pyramid and the CRF inference) and its claimed localization performance is questionable because a standard dataset was used [22], but with a proprietary evaluation metric.

An image operator Stroke Width Transform (SWT) was introduced by Epshtein et al. [10]. The SWT method finds edges using Canny detector [3] and then estimates stroke width for each pixel in the image (see Figure 3). Connected component algorithm is then applied to form pixels with similar stroke width into character candidates, which are merged into text blocks using several heuristic rules. The biggest limitation of the method is its dependency on successful edge detection which is likely to fail on blurred or low-contrast images. The method was further improved by Yao et al. [45] where the heuristic rules for character candidate detection and text block formation are replaced by trained classifiers with rotationally-invariant features.

(a) (b)

Figure 3: Stroke Width Transform (SWT). (a) the original image converted to grey-scale. (b) stroke width estimation for each pixel. *Images adopted from [10]*

2.1.2 Sliding window methods

Methods described in this section use a texture-based approach for text detection. They use a *sliding window* which is moved over the whole image and presence of text is tested for each position. While these methods are generally more robust to noise in the image, their computation complexity is high because of the need to search with many rectangles of different sizes, aspect ratios and potentially rotations. Additionally, support for slanted or perspectively distorted text is limited and sliding window methods do not always provide text segmentation which can be used for character recognition.

Chen and Yuille [4] use AdaBoost classifier [38] combining mean intensity features, intensity variance features, derivative features, histogram features and features based on edge linking. A variant of Niblack's adaptive binarization algorithm [31] is then used to obtain segmentation. The method is computationally expensive (it takes 3 seconds to process a 2MPix image), it requires manual segmentation of many subwindows for training purposes (see Figure 4) and its localization performance is not clear (standard evaluation protocol was not used) - however based on the outputs provided the method seems to overestimate area of the text.

The method [4] was improved by Pan et. al [34] by incorporating a combination Histogram Of Gradient (HOG) and multi-scale Local Binary Pattern feature in the text detection stage. Furthermore, a Markov Random Field (MRF) is employed to to group segmented characters into words, in contrast to the heuristic rules applied in [4]. The method claims better localization performance than the winner of ICDAR 2005 Text Locating Competition [21], which is somewhat inaccurate because different evaluation units were used (words vs. lines of text). Additionally, the method suffers from high compu-

Figure 4: Positive training data samples used for sliding window classifier training. *Image adopted from [4]*

tational complexity (average processing time 1.5s on a 1MPix image).

More recently, Lee et al. [15] further improved the approach by incorporating more discriminative but also more computationally expensive features, which slightly improved text localization performance, but significantly slowed down the method (processing time is several minutes per image).

Coates et al. [6] use unsupervised machine-learning techniques independently for character detection and recognition. A 32-by-32 pixel window is shifted over the image in multiple scales and each patch is classified using a linear SVM classifier as text or non-text. Features used by the classifier are generated automatically in the training stage using a variant of the K-means algorithm. Cropped characters are recognized by resizing them into a fixed 32-by-32 pixel window and applying the same process. The method however does not provide end-to-end text recognition (characters are cropped manually) and it requires significant volumes of training data (order of 10^4 training samples) for both character detection and recognition, which is impractical.

2.2 Lexicon-based word spotting

Due to the complexity of the general text localization and recognition problem, some methods focus on a more constrained scenario when a relatively small lexicon of words is given with each image and the aim is to localize only the words present in the lexicon. This constrained scenario still has many interesting applications, such as local navigation system for the blind (where possible names of local businesses in the area are known based on current GPS position, but their exact location needs to be determined through a vision system).

Wang and Belongie [43] detect and recognize individual characters using a multi-scale sliding window approach. A Histograms of Oriented Gradient

(HOG) features are calculated for each window position and a nearest neighbor classifier is used to measure distance between the patch and all character templates in all classes. Then, each word in the lexicon is considered and the cost of its character configuration is estimated using pictorial structures [11] that penalize disagreement with recognized labels and layout deformation. The method was further improved in [42], where the nearest neighbor classifier was replaced by Random Ferns [2] and an SVM classifier [7] is applied for word re-scoring in order to incorporate higher-order features of word configuration (e.g. standard deviation of character spacing). It is demonstrated that the method performs well on noisy or otherwise distorted images, however the accuracy significantly decreases with increasing size of the lexicon.

2.3 Cropped text recognition

Several methods that focus solely on text recognition were published. The methods assume that text had been already localized in a previous stage and they aim to find string representation of characters/words that were cut out from the original image. This assumption is an over-simplification of the text localization and recognition problem, because in some cases it is not possible to determine word boundaries without knowing its content, but the methods provide useful guidelines that can be applied in methods for general text localization and recognition.

Cropped character recognition was studied by several authors [46, 19, 17, 8, 30]. The methods differ mostly in the color space and features used; the most promising results were achieved with features based on Histogram Of Gradients ([8, 30]).

Mancas-Thillou and Gosselin [23] study cropped word binarization by clustering in RGB color space using Euclidian and Cosine distance. Kim et al. [12] exploit user interaction on a mobile device to find initial location of text and then color clustering in HCL color space [37] is used to find initial candidate regions. The regions are then expanded in horizontal direction using a set of heuristic rules to obtain blocks of text.

Weinman et al. [44] combine lexicon, similarity and appearance information into a joint model and use Sparse Belief Propagation [33] to infer the most probable string content. Similarly, Mishra et al. [25] use a joint CRF model [14] to combine individual character detection results with a language model (lexicon).

2.4 Correcting text recognition

Like printed document recognition (OCR), scene text recognition can benefit from output post-processing to improve its accuracy - this process is however more difficult because scene text is often a very short snippet without any relation to any of the surrounding text. Beaufort and Mancas-Thillou [1] use a weighted Finite State Machine to find the most probable sequence of characters from a list of OCR hypotheses. Donoser et al. [9] detect individual characters as Maximally Stable Extremal Regions [24], classify them using cross-correlation with training samples and then use standard search engines' scoring systems to select the most probable hypothesis and correct any misspellings.

3 Our Work

We have proposed an end-to-end real-time scene text localization and recognition method which achieves state-of-the-art results on standard datasets (we consider a text recognition method real-time if the processing time is comparable with the time it would take a human to read the text).

In [26] (Appendix A), a novel framework for end-to-end text localization and recognition that departs from a strict feed-forward pipeline and replaces it by a hypotheses-verification framework simultaneously processing multiple text line hypotheses was presented. We were the first method to report both text detection and recognition results on the standard and rather challenging ICDAR 2003 dataset [22].

In [27] (Appendix B), we investigate the estimation of text line parameters and exploited them to construct an efficient text line formation algorithm. The presented method was the first one to use top- and bottom-lines as constraints in the text line formation process.

In [28] (Appendix C), an efficient grouping of characters through pruned exhaustive search was introduced. Additionally, it was demonstrated that the grouping stage plays a key part in the whole text localization process and that false positives can be significantly reduced by incorporating higher-order features. Finally, MSER lattice [24] induced by the inclusion relation was exploited to select an optimal segmentation.

In [29] (Appendix D), a real-time method which achieves state-of-the-art results on standard datasets [21, 43] was presented. The method evaluates all possible segmentations of a character using novel features calculated with $O(1)$ complexity per region tested and a highly efficient exhaustive search with feedback loops is then applied to group ERs into words and to select

the most probable character segmentation.

4 Goals of the Thesis

The main goal of the thesis is to propose a general end-to-end scene text localization and recognition method that will achieve results comparable to humans on standard datasets.

Even though the method currently achieves state-of-the-art results, there are still many interesting open problems and space for improvements (the overall recall on the standard dataset [39] is only 65%):

- **Hierarchical textual information.** The method (and all the previously published methods) output an unorganized set of words, which is insufficient for practical applications where the order of words is important (e.g. an assistant application for blind people). Understanding relations amongst individual words can also improve recognition accuracy because n-grams can be exploited to correct OCR output.

- **A character is not a connected component.** A current limitation of the method is an assumption that each character is a separate connected component. The assumption can be violated in two ways: either a character consists of multiple connected components (e.g. segment displays or characters made of dots) or several characters (or a whole word) are one connected component (e.g. logos, Arabic or Indian scripts).

- **Multi-language and multi-script localization and recognition.** The method is currently limited to texts in Latin alphabet (without accents) in one language. This needs to be extended to multiple languages and multiple scripts (e.g. Cyrillic script, Kannada script, etc.).

- **Robustness to rotation, perspective distortion and noise.** The method's robustness to rotation is small (approx. $\pm 20°$) and vertical text is not detected at all. Noise introduced by JPEG compression has a negative impact on the overall recall.

A A method for text localization and recognition in real-world images

A reprint of our article *A method for text localization and recognition in real-world images* presented at the Tenth Asian Conference on Computer Vision (ACCV 2010).

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 [1–4]. 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, need for geometric normalization compensating 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, 6–8] 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 individual success rates at 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, 9–11, 4], individual character recognition [12, 13] or

reading text from segmented areas of images [14]. Whilst they achieved promising results on the 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).

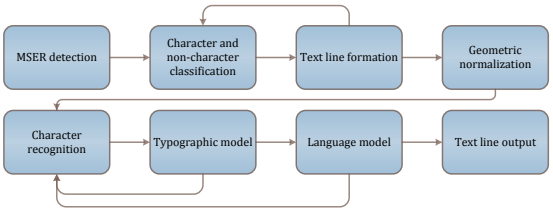


Fig. 1. Stages of the proposed method (incl. feedback loops for hypotheses verification)

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. 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.

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.

The proposed method is also novel in avoiding a pipeline architecture with a sequence of fixed decisions. The method works with multiple hypotheses at each stage of the processing (text localization, character segmentation, text line formation), revisits early steps in a hypothesis-verify framework and leaves the decision about the most probable hypothesis to the final stage, 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 the sequence s. Two sequences are considered adjacent, if first one was created 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 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 a more advantageous approach to describe the implementation of the proposed 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.

Accordingly, text recognition is represented as a labeling of each such path p with a textual content $\mathbf{y} = y_1 y_2 \dots y_l : y_i \in \mathcal{A}$.

3 Text localization and recognition

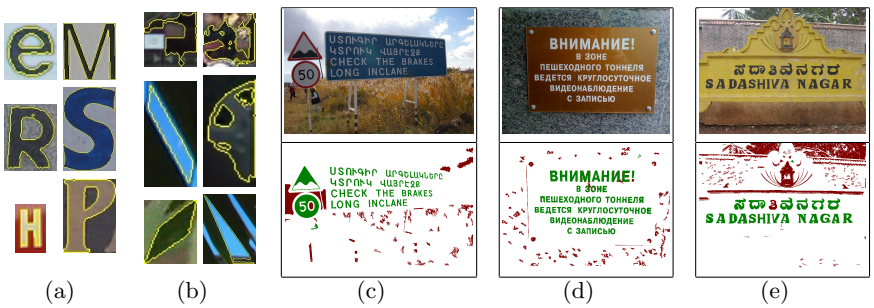
3.1 MSER detection

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 as needed (see Section 3.3).

3.2 Character and non-character classification

In the next stage, 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.



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

Fig. 3. Character and non-character classifier of MSERs: (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

The features used by the classifier (see Table 1) are scale invariant so that characters of all sizes are detected, but they are not rotation invariant, which

implies that characters of different rotations had to be included in the training set.

Once text lines are built (see Section 3.3) the initial decisions made by the classifier are rectified using a feed-back loop based on inferred values of the hidden parameters of the text lines (character height, character spacing, etc.). Thanks to the feed-back loop, the classification error does not have any 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 characters and 1396 non-characters obtained by manually annotating MSERs extracted from real-world images downloaded from an on-line photo storage[1]. The classification error on the training set obtained by cross-validation is 5.6%. The training set is relatively small and certainly does not contain all possible fonts, scripts or even characters, but it was experimentally proven that extending the training set with more examples does not bring significant improvement in the classification success rate. Based on this experiment, it can be assumed that features used by the character classifier are invariant to various 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

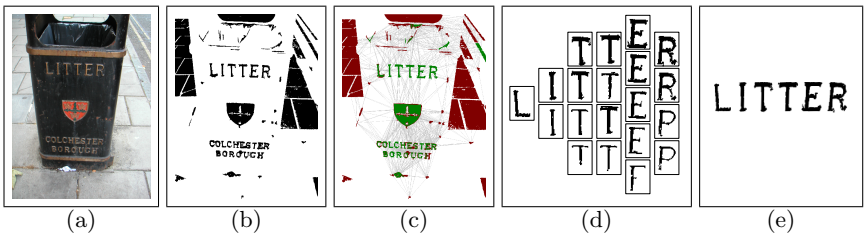
3.3 Text line hypothesis formation

It has been observed that 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 trained classifier, 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 ICDAR 2003 Train set [20] was used to create a training set for the SVM classifier with the polynomial kernel.

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.

[1] http://www.flickr.com



(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.

Each path p is built in the following manner: The top-left 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 $v \in V(\mathbf{G})$ 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.

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]

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 nodes of the path are marked as closed and next open 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 fitting 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. Least-Median Squares (LMS) method is used to create the approximation - if the path is shorter than 5 vertices, only straight bottom line is allowed; if the path is longer, we allow the bottom line to be slightly curved by fitting the bottom points with a parabola (see Figure 11, bottom-left).

3.4 Geometric normalization

Perspective distortion often influences the text in real-world images. The distortion is rectified prior to character recognition as all characters are trained in the frontoparallel view.

Orientation of a camera to a plane with text in 3D space can be represented as a homography with a transformation matrix $\mathbf{H}$, which can be 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. The rotation θ can be easily calculated from 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 method of Myers et al. [21] is used to estimate the sought parameters. 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.

In order to find the top line, the text block is rotated in the range of ± 3 degrees by 0.2 degree increments from detected text line orientation. 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, with the exception that 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 mask of the MSER to a fixed-sized matrix of 35×35 pixel, while retaining the centroid of the region and aspect ratio [22]. Next, a chain-code of the character perimeter is generated and for each chain-code direction the corresponding pixels are extracted into a separate bitmap. On each bitmap a regular 7×7 Gaussian masks are evenly placed to generate 25 features. In total, 25 features $\times$ 8 directions generate 200 features for each MSER mask.

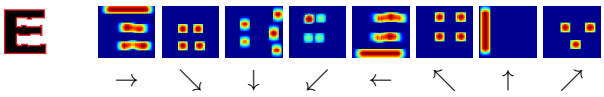


Fig. 6. Character recognition features: Input character (left). Features of the chain-code bitmap for each direction (right).

A SVM classifier with Radial Basis Function (RBF) kernel was used. 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.

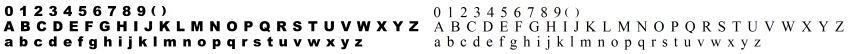
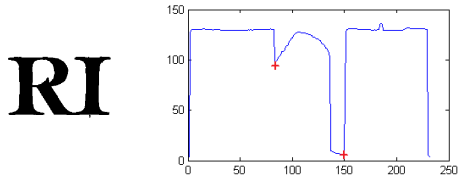


Fig. 7. Synthetic training samples of character classifier. No "real world" training samples were used.

If 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 local minimum of distance between top and bottom pixel in a column (see Figure 8). Each combination of splitting is then evaluated in the context of surrounding letters using a feed-back loop (see Section 3.7) and the hypothesis with the highest score according to the language model is selected.

3.6 The typographic model

A feed-back loop for character recognition was introduced in the proposed method, because 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



Split regions	Recognized text
	ItI
	Iu
	RI
	N

(a) (b) (c)

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

unmistakably 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 the 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 rs 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	0.0528
	LITTFR	0.0356
	LITTER	**0.0814**
	LITTFP	0.0168

4 Experiments

4.1 Chars74K dataset

The performance of the proposed method was evaluated on the *Chars74K*[2] 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.

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

[2] 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.

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 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[3], which contains 5370 letters and 1106 words in 249 pictures. The same configuration as in the previous experiment (see Section 4.1) was used. The dictionaries supplied with the ICDAR 2003 dataset were not used in order to evaluate generic performance of the method. 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.

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

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)

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. Example of the ICDAR 2003 images where the proposed method fails to localize the text

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, 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).

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** (2004) 87–99
3. Ezaki, N.: Text detection from natural scene images: towards a system for visually impaired persons. In: In Int. Conf. on Pattern Recognition. (2004) 683–686
4. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR '10: 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. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **2** (2004) 366–373
7. Gao, J., Yang, J.: An adaptive algorithm for text detection from natural scenes. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **2** (2001) 84
8. Jain, A.K., Yu, B.: Automatic text location in images and video frames. Pattern Recognition, International Conference on **2** (1998) 1497
9. Pan, Y.F., Hou, X., Liu, C.L.: A robust system to detect and localize texts in natural scene images. Document Analysis Systems, IAPR International Workshop on **0** (2008) 35–42
10. Kim, E., Lee, S., Kim, J.: Scene text extraction using focus of mobile camera. Document Analysis and Recognition, International Conference on **0** (2009) 166–170
11. Pan, Y.F., Hou, X., Liu, C.L.: Text localization in natural scene images based on conditional random field. In: ICDAR '09: Proc. of the 2009 10th International Conference on Document Analysis and Recognition. (2009) 6–10
12. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. VISAPP **05-08 February 2009** (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. (2005) 167–171
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** (2009) 1733–1746
15. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22** (2004) 761–767
16. Matas, J., Zimmermann, K.: A new class of learnable detectors for categorisation. In: Proc. of the 14th Scandinavian Conference on Image Analysis. (2005) 541–550
17. Nistér, D., Stewénius, H.: Linear time maximally stable extremal regions. In: Proc. of the 10th European Conference on Computer Vision. (2008) 183–196
18. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines. Cambridge University Press (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** (2001) 181–201
20. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: ICDAR '03: Proc. of the 7th International Conference on Document Analysis and Recognition. (2003) 682
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** (2005) 147–158
22. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognition **37** (2004) 265 – 279
23. Lucas, S.M.: Text locating competition results. Document Analysis and Recognition, International Conference on **0** (2005) 80–85

B Estimating hidden parameters for text localization and recognition

A reprint of our article *Estimating hidden parameters for text localization and recognition* presented at the Sixteenth Computer Vision Winter Workshop (CVWW 2011).

16^{th} *Computer Vision Winter Workshop*
Andreas Wendel, Sabine Sternig, Martin Godec (eds.)
Mitterberg, Austria, February 2-4, 2011

Estimating hidden parameters for text localization and recognition

Lukáš Neumann
neumalu1@cmp.felk.cvut.cz

Jiří Matas
matas@cmp.felk.cvut.cz

Center for Machine Perception
Department of Cybernetics
Czech Technical University
Prague, Czech Republic

Abstract. *A new method for text line formation for text localization and recognition is proposed. The method exhaustively enumerates short sequences of character regions in order to infer values of hidden text line parameters (such as text direction) and applies the parameters to efficiently limit the search space for longer sequences. The exhaustive enumeration of short sequences is achieved by finding all character region triplets that fulfill constraints of textual content, which keeps the proposed method efficient yet still capable to perform a robust estimation of the hidden parameters in order to correctly initialize the search. The method is applied to character regions which are detected as Maximally Stable Extremal Regions (MSERs).*

The performance of the method is evaluated on the standard ICDAR 2003 dataset, where the method outperforms (precision 0.60*, recall* 0.60*) a previously published method for text line formation of MSERs.*

1. Introduction

Text localization and recognition in images of real-world scenes is still an open problem, which has been receiving significant attention in the last decade [12, 1, 5, 4, 10, 3]. In contrast to text recognition in documents, which is satisfactorily addressed by state-of-the-art OCR systems [6], no efficient method for scene text localization and recognition has been yet published.

Methods for text localization are based on two approaches: sliding windows and connected component analysis. The methods based on sliding windows [2] are more robust to noise, but they have high computational complexity (scanning whole image with windows of multiple sizes is required) and they cannot detect slanted or perspectively distorted text. That is why methods based on individual region detection and subsequent connected component analysis are getting more attention in the text localization community [5, 4, 10]. On the most cited dataset (ICDAR 2003 [8]) the methods based on connected component analysis achieve state-of-the-art results in text localization [11].

In this paper, we present a text line formation method, which groups Maximally Stable Extremal Regions (MSERs) [9] representing characters into text lines. The main contribution of this work is an ability to exhaustively enumerate short sequences of character regions in order to infer values of hidden text line parameters (such as text direction) and subsequently applying the parameters to efficiently limit the search space for longer sequences. The exhaustive enumeration of short sequences is achieved by finding all character region triplets that fulfill constraints of textual content, which keeps the proposed method efficient yet still capable to perform a robust estimation of the hidden parameters in order to correctly initialize the search. The method was evaluated using the hypotheses-verification framework for text localization and recognition published by Neumann and Matas [10], where the heuristic text line formation stage was replaced by the proposed method.

The rest of the document is structured as follows: In Section 2, hidden text line parameters used by the proposed method are defined. Section 3 describes the proposed method for text line formation. Performance evaluation of the proposed method is pre-

sented in Section 4. The paper is concluded in Section 5.

2. Hidden text line parameters

It can be observed that text in real-world images follows a certain structure. The structure is not as strict as in the case of text in printed documents, but it is possible to make certain observations at least on the level of individual words; text parameters such as character height, character color, spacing between individual characters have only limited number of distinct values inside a single word. Moreover each word (and possibly more than one word) has an implied direction in which all characters are laid out.

In this paper, we refer to all such parameters as *hidden text line parameters* (or just *hidden parameters*). The initial values of the hidden parameters are obtained by exhaustively enumerating all region triplets and then the inferred values are used to limit the search space during next steps of the text formation. The hidden text line parameters used by the proposed method are height ratio (Section 2.1), centroid angle (Section 2.2) and text direction (Section 2.3).

2.1. Height ratio

The height of two following letters in a word is constrained to a limited interval. In order to express this relation, the height ratio hr between two characters c^1 and c^2 is introduced as

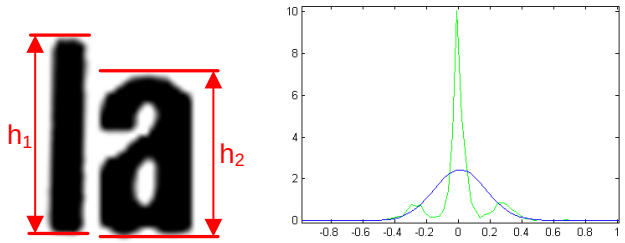
$$\mathsf{hr}(c^1, c^2) = \log \frac{h_1}{h_2} = \log \frac{c_b^1 - c_t^1}{c_b^2 - c_t^2} \qquad (1)$$

where c_t^i and c_b^i denote top and bottom co-ordinate of a bounding box of the character c (see Figure 1a). The measurement is scale invariant, but it is not rotation invariant, which implies that various rotations had to be included in the training set.

Figure 1b depicts the normalized histogram of height ratio values in the training set and their inferred approximation using a Gaussian Mixture Model.

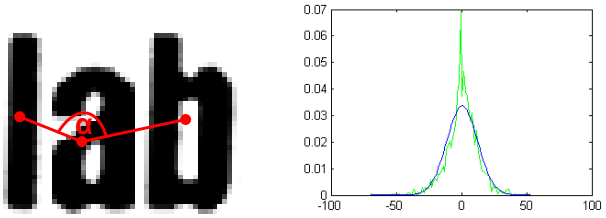
2.2. Centroid angle

Given a sequence of three following letters in a word, the angle between lines connecting their centroids (see Figure 2a) is also constrained to a limited interval. The centroid angle ca of three characters c^1, c^2 and c^3 is defined as



(a) (b)

Figure 1. Height ratio. (a) Measurement example. (b) Normalized histogram (green) and inferred Gaussian Mixture Model M_{hr} (blue)



(a) (b)

Figure 2. Centroid angle. (a) Measurement example. (b) Normalized histogram (right, green) and inferred Gaussian Mixture Model M_{ca} (blue)

$$\mathsf{ca}(c^1, c^2, c^3) = \left| \arctan\left(\frac{c_{cy}^1 - c_{cy}^2}{c_{cx}^1 - c_{cx}^2}\right) - \arctan\left(\frac{c_{cy}^2 - c_{cy}^3}{c_{cx}^2 - c_{cx}^3}\right) \right| \qquad (2)$$

where c_{cx}^i (c_{cy}^i) denotes horizontal respectively vertical co-ordinate of a centroid of the character c^i. The measurement is both scale and rotation invariant.

Figure 2b depicts the normalized histogram of centroid angle values in the training set and their inferred approximation using a Gaussian Mixture Model.

2.3. Text direction

The structure of text in real-world images exhibits higher-order properties, which cannot be fully captured by measurements which are defined only using pairs or triplets of individual characters (such as the parameters in Sections 2.1 and 2.2).

In this paper we introduce a set of parameters called *text direction* to capture higher-order structure of text, which exploits an observation that the top and bottom boundaries of individual characters in a word can be fitted by a line. Depending on which letters form the word, each word has either 1 or 2 top lines (see Figure 3), depending whether only upper-case or both upper-case and lower-case letters are

present in the word. Let $t_1(x)$ and $t_2(x)$ denote vertical position of first respectively second top line at point x. The same observation applies to the bottom lines where either 1 or 2 lines are present, depending whether underline characters such as "y" or "g" are present or not. Let $b_1(x)$ and $b_2(x)$ again denote vertical position of the bottom lines at point x. *Text direction* T is then defined as quaternion (t_1, t_2, b_1, b_2).

Given a text direction T, *text direction distance* of a character c is defined as

$$d(c,T) = \max\Big(\min(|t_1(c_l) - c_t|, |t_2(c_l) - c_t|), \min(|b_1(c_l) - c_b|, |b_2(c_l) - c_b|)\Big) \quad (3)$$

where c_t, c_l and c_b denote top, left and bottom coordinate of a bounding box of the character c.

Mutual position of the lines is not arbitrary either. An assumption was made that these lines are parallel, because height of individual characters in a single word is assumed to be constant and effects caused by perspective distortion in a single word are marginal. Let $D(a(x), b(x)) = |a(x) - b(x)|$ denote vertical distance between lines a and b at horizontal co-ordinate x. Since it was assumed that the lines are parallel, the distance D does not depend on the horizontal position and we can simply write $D(a, b)$ for distance between lines a and b.

In order to express the constraints for mutual vertical distance of the lines, a height of a top bend h_t, a middle bend h_m and a bottom bend h_b is defined (see Figure 3) as

$$h_t(T) = D(t_1, t_2) \quad (4)$$

$$h_m(T) = D(\max(t_1, t_2), \min(b_1, b_2)) \quad (5)$$

$$h_b(T) = D(b_1, b_2) \quad (6)$$

In order to make the text direction parameters scale invariant, they are normalized using a maximal height of a character in the word $h_{\max}$:

$$\bar{d}(c) = \frac{d(c)}{h_{\max}} \quad (7)$$

$$\bar{h}_t(T) = \frac{h_t(T)}{h_{\max}} \quad (8)$$

$$\bar{h}_m(T) = \frac{h_m(T)}{h_{\max}} \quad (9)$$

$$\bar{h}_b(T) = \frac{h_b(T)}{h_{\max}} \quad (10)$$

As shown in Figure 4 the variance of text direction distance $\bar{d}(c)$ measured on the training set is relatively small, which suggests that this parameter can be used as a feature to distinguish between textual and non-textual structures.

Figure 3. Text direction - top lines (red) and bottom lines (green)

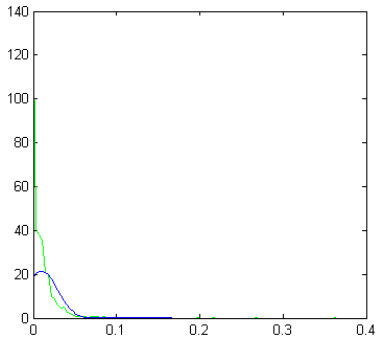


Figure 4. Text direction distance $\bar{d}(c)$ - histogram (green) and inferred Gaussian Mixture Model M_d (blue)

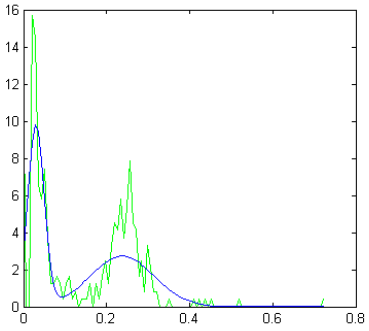


Figure 5. Top band height $\bar{h}_t$ - histogram (green) and inferred Gaussian Mixture Model M_{tb} (blue)

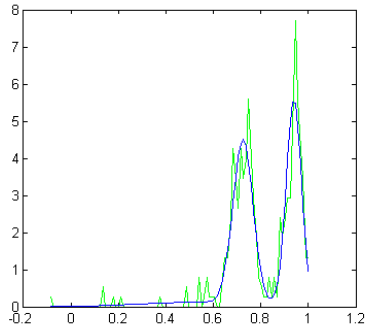


Figure 6. Middle band height $\bar{m}_t$ - histogram (green) and inferred Gaussian Mixture Model M_{mb} (blue)

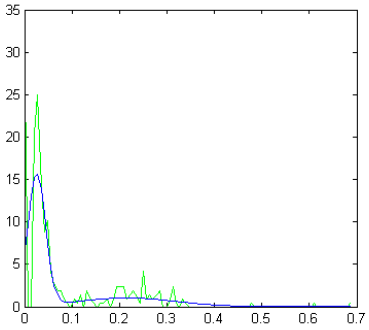


Figure 7. Bottom band height $\bar{b}_t$ - histogram (green) and inferred Gaussian Mixture Model M_{bb} (blue)

```
Procedure la(cc)
 tp := top points of all chars in cc
 bp := bottom points of all chars in cc
 ap := fit bp by a line using Least-Median Squares
 k := tangent of ap

 t1,t2 := fit(tp, k)
 b1,b2 := fit(bp, k)
 T := (t1, t2, b1, b2)
 return T

Procedure fit(points, k)
 bestError := Inf
 for each p,q in points
  line1 := line through p with tangent k
  line2 := line through q with tangent k

  error := 0
  for each r in points
   dist := (min(line1(r[x]),line2(r[x]))-r[y])^2
   error := error + dist

  if error < bestError
   bestError := error
   l1 := line1
   l2 := line2

 return (l1, l2)
```

Figure 8. Pseudo-code of the text direction approximation procedure $\mathsf{la}(cc)$

In order to obtain the text direction T from a sequence of characters $cc = c^1, c^2 \dots c^n$ a procedure $\mathsf{la}(cc)$ is introduced (see Figure 8). The example output of the procedure is shown in Figure 9.

3. Text line formation

3.1. Region graph

Individual characters are obtained by detecting Maximally Stable Extremal Regions (MSERs) [9] and then including only the MSERs which are classified as characters using a trained classifier, as proposed by Neumann and Matas [10].

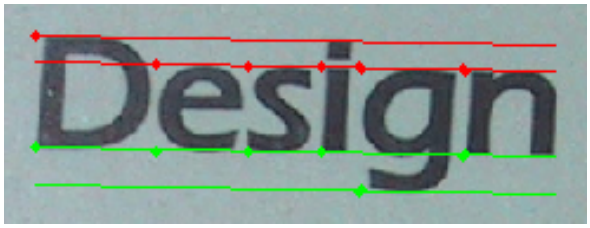


Figure 9. Sequence of characters cc with marked top (red) and bottom (green) points and text direction (top lines - red, bottom lines - green) obtained using the procedure $\mathsf{la}(cc)$

Figure 10. Region graph (initial configuration without any edge labeling)

Let $G = (V, E)$ denote the region graph. The set of vertices V corresponds to the set of character MSERs found in the image. The set of edges E is formed in the following matter: For each vertex, edges to 3 nearest neighboring vertices to the right are created (whilst excluding edges whose centroid angle α is above $40°$). The distance between two vertices is measured as the distance between their centroids. Figure 10 shows an example of such a graph.

3.2. Graph energy

Let $f : E \to \{0, 1\}$ denote a configuration of the region graph G. The text localization task is formulated as finding the best configuration f^* of given graph G such that graph energy $\mathcal{E}(G, f)$ is minimal:

$$f^* = \underset{f}{\operatorname{argmin}}\, \mathcal{E}(G, f) \tag{11}$$

The energy $\mathcal{E}$ is composed of the following weighted components

$$\begin{aligned}\mathcal{E}(G, f) = \alpha_1 \mathcal{E}_{hr}(G, f) + \alpha_2 \mathcal{E}_{ca}(G, f) \\ + \alpha_3 \mathcal{E}_{d}(G, f) + \alpha_4 \mathcal{E}_{la}(G, f)\end{aligned} \tag{12}$$

where $\mathcal{E}_{hr}$ denotes energy of character height ratios (see Section 2.1), $\mathcal{E}_{ca}$ denotes energy of character centroid angles (see Section 2.2) and $\mathcal{E}_d$ ($\mathcal{E}_{la}$) denotes energy of text direction distances and energy of line approximation respectively (see Section 2.3). Coefficients α_i then denote non-negative weights, which in our setup were all set to 1 in order to give each energy an identical weight. The individual energy components are defined using a Gaussian Mixture Model (GMM) approximation, which was created using the training dataset (as shown in Figures 1, 2, 4, 5, 6 and 7).

Given a Gaussian Mixture Model M obtained from training data

$$f(x) = \sum_{i=1}^{n} \alpha_i \mathcal{N}_{\mu_i,\sigma_i}(x) = \sum_{i=1}^{n} \alpha_i \mathcal{N}_M(x) \quad (13)$$

the energy $\mathcal{L}_M(x)$ for corresponding model M at point x is defined as

$$\mathcal{L}_M(x) = \min \left\{ \left(\frac{\mu_i - x}{\sigma_i} \right)^2 : i = 1 \dots n \right\} - \theta \quad (14)$$

where θ denotes a threshold parameter defining what square distance from mean value is considered acceptable. In our setup the value θ was set so that 95% values from training data is accepted.

Let E' denote a subset of edges $\{e \in E \,|\, f(e) = 1\}$ of the graph G and let $C(G, f)$ denote a set of strongly connected components of the graph G when taking into account only edges in E'.

The energy of character height ratios $\mathcal{E}_{hr}(G, f)$ is defined as

$$\mathcal{E}_{hr}(G, f) = \sum_{e \in E'} \mathcal{L}_{M_{hr}} \left(\mathsf{hr}\left(e_b, e_e\right) \right) \quad (15)$$

where e_b (e_e) denotes a vertex where the edge e begins (ends).

The energy of character centroid angles $\mathcal{E}_{ca}(G, f)$ is defined as

$$\mathcal{E}_{ca}(G, f) = \sum_{\substack{e^1, e^2 \in E' \\ e^1_e = e^2_b}} \mathcal{L}_{M_{ca}}(\mathsf{ca}(e^1_b, e^1_e, e^2_e)) \quad (16)$$

where again e^i_b (e^i_e) denotes a vertex where the edge e^i begins (ends).

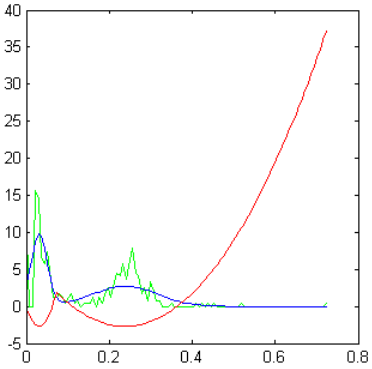


Figure 11. Normalized histogram of training data (green), inferred Gaussian Mixture Model M (blue) and corresponding energy function $\mathcal{L}_M$ (red)

The energy of text direction distances $\mathcal{E}_d(G, f)$ and energy of line approximation $\mathcal{E}_{la}$ are defined as

$$\mathcal{E}_d(G, f) = \sum_{cc \in C(G,f)} \sum_{c \in cc} \mathcal{L}_{M_d} \left(\frac{d(c, \tau)}{h_{\max}} \right) \quad (17)$$

$$\mathcal{E}_{la}(G, f) = \sum_{cc \in C(G,f)} \max \left\{ \mathcal{L}_{M_{tb}} \left(\frac{h_t(\tau)}{h_{\max}} \right), \mathcal{L}_{M_{mb}} \left(\frac{h_m(\tau)}{h_{\max}} \right), \mathcal{L}_{M_{bb}} \left(\frac{h_b(\tau)}{h_{\max}} \right) \right\} \quad (18)$$

$$\tau = \mathsf{la}(cc),\ h_{\max} = \max_{c' \in cc}(c'_b - c'_t)$$

3.3. Building region sequences

Region sequences are iteratively built by altering the graph configuration f in order to minimize the energy of the graph $\mathcal{E}(G, f)$. In each step the procedure `test` compares energy of newly created graph configuration f' to the best energy found so far and if a lower energy is found, the current configuration f is updated.

The method starts by enumerating all region triplets, taking only the acceptable triplets (the ones which decrease the graph energy $E(G, f)$) and thus initializing values of text line hidden parameters. Then, single regions are enumerated and the hidden text line parameters are used to efficiently prune the search space. As a last step the method tries to disconnect regions based on the inferred parameters of the whole line of text, because some regions might have been connected in the early stage as a result of inaccurate hidden parameters estimation on short sequences. The process is outlined in Figure 12, a result of the process is shown in Figure 13.

```
Procedure findBestConfiguration (G)
 f := (0,0, ... 0)
 E := 0
 { Connecting triplets of regions to obtain
   initial values of hidden parameters }
 for each subsequent pair of edges e,e' in G
  f' := f
  f'(e, e') = 1
  (E, f) := test(E, f, f')

 { Connecting single regions }
 for each edge e in G
  f' := f
  f'(e) := 1
  (E, f) := test(E, f, f')

 { Trying to disconnect pairs of nodes }
 for each edge e in G
  f' := f
  f'(e) := 0
  (E, f) := test(E, f, f')

 return f

Procedure test(E, f, f')
 E' = calculateEnergy(f')
 if E' < E
  E := E'
  f := f'

 return (E, f)
```

Figure 12. Pseudo-code of finding the best region graph configuration f in the region sequences building process

Figure 13. Region graph and its edge labeling corresponding to the best configuration (edges of the graph $f(e) = 1$ marked green, $f(e) = 0$ marked red)

4. Experiments

The method was evaluated using the hypothesis-verification framework proposed by Neumann and Matas [10] and replacing the heuristics text formation stage by the proposed method. The standard and most cited ICDAR 2003 Robust Reading Competition dataset[1] [8] was used for performance evaluation. The Train set was used to obtain the method parameters and an independent Test set was used to evaluate the performance. In total the ICDAR 2003 Test set contains 5370 letters and 1106 words in 249 pictures.

method	precision	recall	f
Pen et. al [11]	0.67	0.71	0.69
Zhang et. al [13]	0.73	0.62	0.67
Epshtein et. al [4]	0.73	0.60	0.66
Hinnerk Becker [7]	0.62	0.67	0.62
proposed method	**0.60**	**0.60**	**0.60**
Alex Chen [7]	0.60	0.60	0.58
Neumann and Matas [10]	0.59	0.55	0.57
Ashida [8]	0.55	0.46	0.50
HWDavid [8]	0.44	0.46	0.45
Wolf [8]	0.30	0.44	0.35
Qiang Zhu [7]	0.33	0.40	0.33
Jisoo Kim [7]	0.22	0.28	0.22
Nobuo Ezaki [7]	0.18	0.36	0.22
Todoran [8]	0.19	0.18	0.18

Table 1. Text localization results on the ICDAR 2003 dataset

Applying the evaluation protocol defined in [8], the proposed method achieved precision of 0.60 and recall of 0.60, which gives f-measure of 0.60. Figure 14 shows examples of text localization and recognition on the ICDAR 2003 dataset.

5. Conclusions

A novel method for text line formation was proposed. The method uses the hidden parameters of the text line (such as text direction) to group Maximally Stable Extremal Regions (MSERs) into lines of text. The exhaustive enumeration of short sequences is achieved by finding all character region triplets that fulfill constraints of textual content, which keeps the proposed method efficient yet still capable to perform a robust estimation of the hidden parameters in order to correctly initialize the search.

The proposed method was evaluated on the standard ICDAR 2003 dataset using the standard evaluation protocol [8], where it outperforms the method for forming text lines of Neumann and Matas [10] (f-measure is increased from 0.57 to 0.60). The method is still behind the state-of-the-art method for text localization (Pen et al. [11], f-measure 0.69), but the text localization results have to be interpreted carefully as there are known problems with the evaluation

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

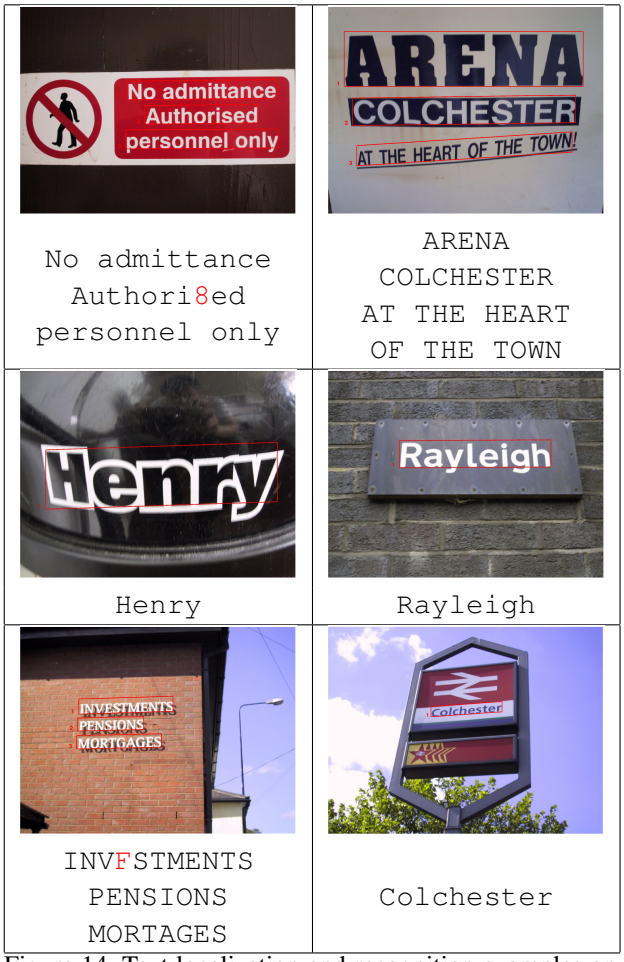


Figure 14. Text localization and recognition examples on the ICDAR 2003 dataset.

protocol and ground truth of the ICDAR 2003 dataset [7, 10]. The proposed method aims to solve the complete problem of text detection and recognition (see Figure 14), however all the methods superior in text localization performance [11, 13, 4, 7] aim only to solve one part of the problem and thus direct comparison cannot be made.

Most frequent problems of the proposed method is unsupported text line structure (Figure 15a), symbols or pictographs placed close to text lines (Figure 15b), letters not detected as individual regions (Figure 15c) and false positives caused by repetitive textures with a text-like spacial structure (Figure 15d).

Acknowledgement. The authors were supported by Czech Government reseach program MSM6840770038.

(a) (b)

(c) (d)

Figure 15. Problems of the proposed method. (a) Unsupported text line structure. (b) Pictographs placed close to text lines. (c) Letters not detected as individual regions. (d) False positives caused by repetitive textures with a text-like spacial structure

References

[1] X. Chen, J. Yang, J. Zhang, and A. Waibel. Automatic Detection and Recognition of Signs From Natural Scenes. *IEEE Trans. on Image Processing*, 13:87–99, Jan. 2004.

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:366–373, 2004.

[3] M. Donoser, H. Bischof, and S. Wagner. Using web search engines to improve text recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4, 2008.

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR '10: Proc. of the 2010 Conference on Computer Vision and Pattern Recognition*, 2010.

[5] N. Ezaki. Text detection from natural scene images: towards a system for visually impaired persons. In *In Int. Conf. on Pattern Recognition*, pages 683–686, 2004.

[6] X. Lin. Reliable OCR solution for digital content re-mastering. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Dec. 2001.

[7] S. M. Lucas. Text locating competition results. *Document Analysis and Recognition, International Conference on*, 0:80–85, 2005.

[8] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *ICDAR '03: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, page 682, Washington, DC, USA, 2003. IEEE Computer Society.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable

extremal regions. *Image and Vision Computing*, 22(10):761–767, September 2004.

[10] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *ACCV 2010: Proceedings of the 10th Asian Conference on Computer Vision*, volume IV of *LNCS 6495*, pages 2067–2078, Heidelberg, Germany, November 2010. Springer.

[11] Y.-F. Pan, X. Hou, and C.-L. Liu. Text localization in natural scene images based on conditional random field. In *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 6–10, Washington, DC, USA, 2009. IEEE Computer Society.

[12] V. Wu, R. Manmatha, and E. M. Riseman, Sr. Textfinder: An automatic system to detect and recognize text in images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(11):1224–1229, 1999.

[13] J. Zhang and R. Kasturi. Character energy and link energy-based text extraction in scene images. In R. Kimmel, R. Klette, and A. Sugimoto, editors, *ACCV 2010: Proceedings of the 10th Asian Conference on Computer Vision*, volume II of *LNCS 6495*, pages 832–844, Heidelberg, Germany, November 2010. Springer.

C Text Localization in Real-world Images using Efficiently Pruned Exhaustive Search

A reprint of our article *Text Localization in Real-world Images using Efficiently Pruned Exhaustive Search* presented at the Eleventh International Conference on Document Analysis and Recognition (ICDAR 2011).

Text Localization in Real-world Images using Efficiently Pruned Exhaustive Search

Lukáš Neumann
Centre for Machine Perception, Dept. of Cybernetics
Czech Technical University, Prague, Czech Republic
neumalu1@cmp.felk.cvut.cz

Jiří Matas
Centre for Machine Perception, Dept. of Cybernetics
Czech Technical University, Prague, Czech Republic
matas@cmp.felk.cvut.cz

***Abstract*—An efficient method for text localization and recognition in real-world images is proposed. Thanks to effective pruning, it is able to exhaustively search the space of all character sequences in real time (200ms on a 640×480 image). The method exploits higher-order properties of text such as word text lines. We demonstrate that the grouping stage plays a key role in the text localization performance and that a robust and precise grouping stage is able to compensate errors of the character detector.**

The method includes a novel selector of Maximally Stable Extremal Regions (MSER) which exploits region topology. Experimental validation shows that 95.7% characters in the ICDAR dataset are detected using the novel selector of MSERs with a low sensitivity threshold.

The proposed method was evaluated on the standard ICDAR 2003 dataset where it achieved state-of-the-art results in both text localization and recognition.

***Keywords*-text localization;real-world images;text-in-the-wild**

I. Introduction

Text localization and recognition in real-world images is an open problem, unlike printed document recognition where state-of-the-art systems are able to recognize correctly more than 99% of characters [1]. Applications of text localization and recognition in real-world images range from automatic annotation of image databases based on their textual content (e.g. Flickr or Google Images), assisting the visually impaired to reading labels on businesses in map applications (e.g. Google Street View).

Existing methods for text localization can be categorized into two different groups - methods based on a sliding window and methods based on regions (characters) grouping. Methods in the first category [2], [3] use a window which is moved over the image and the presence of text is estimated on the basis of local image features. While these methods are generally more robust to noise in the image, their computation complexity is high because of the need to search with many rectangles of different sizes, aspect ratios and potentially rotations. Additionally, support for slanted or perspectively distorted text is limited.

The majority of recently published methods for text localization falls into the latter category [4], [5], [6], [7]. The methods differ in their approach to individual character detection, which could be based on edge detection, character energy calculation or extremal region detection. While the methods are paying great attention to individual character detection, grouping of individual characters into words is performed based on heuristics or graph optimization methods and only unary and pairwise constraints are used.

In this paper, a general and efficient method for text localization and recognition is presented, which thanks to effective pruning is able to group character regions by an exhaustive enumeration of all character sequences. The method exploits higher-order properties of text, which cannot be incorporated into standard graph (or hypergraph) optimization methods where only unary or binary features are used. We demonstrate that the grouping stage plays a key role in the text localization performance and that even a character detector with a lower precision is sufficient if the grouping stage is accurate.

As a second contribution, an extended version of Maximally Stable Extremal Regions (MSERs) [8] called MSER++ is introduced. Experimental evaluation shows that 95.7% characters are detected as MSER++, which is a significant improvement over standard MSER (84.0%) as used in our previous method [6].

The rest of the document is structured as follows: In Section II, the proposed method is described, in Section III experimental evaluation is performed and the paper is concluded in Section IV.

II. Text localization

A. Character grouping search space

Let $\mathbf{I}$ denote an image of n pixels and let $P(\mathbf{I})$ denote set of all subregions of the image $\mathbf{I}$. Let s^L denote an arbitrary sequence of non-repeating image subregions $s^L = (r_1, r_2, \ldots, r_L) : r_i \in P(\mathbf{I}),\ r_i \neq r_j\ \forall i, j$ of length L, let $\mathcal{S}^L = \bigcup_{i=1}^{n} s^i$ denote set of all sequences of length L and let $\mathcal{S}$ denote set of all sequences of lengths up to n $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}^i$. Given a verification function $v : \mathcal{S} \rightarrow \{0, 1\}$, the set of estimates (*words*) $\mathcal{E}$ is defined as $\mathcal{E} = \{w \in \mathcal{S} : v(w) = 1\}$. The methods for text localization aim to find an optimal verification function $v^*(s)$ so that f-measure of precision $p = \frac{|\mathcal{E} \cap \mathcal{T}|}{|\mathcal{E}|}$ and recall $r = \frac{|\mathcal{E} \cap \mathcal{T}|}{|\mathcal{T}|}$ is maximized, where $\mathcal{T}$ denotes set of words in the ground truth.

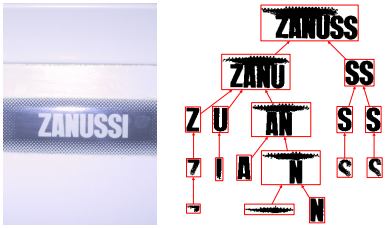


Figure 1. MSER lattice induced by the inclusion relation. Only certain nodes correspond to characters

Table I
INDIVIDUAL CHARACTER DETECTION RATE USING DIFFERENT VARIANTS OF MSER

MSER	root only (%)	complete tree (%)
Greyscale	74.9	93.1
Red channel	72.8	93.2
Green channel	74.9	93.5
Blue channel	72.5	87.1
Combined	84.0	**95.7**

B. Extended Maximally Stable Extremal Regions

The cardinality of the set $\mathcal{S}$ is exponential in the number of pixels in the image, thus it is infeasible to exhaustively search the whole set in order to obtain an optimal solution even if we assume that an optimal verification function $v^*(s)$ exists and can be efficiently calculated. Assuming that each character is a contiguous region of the image $\mathbf{I}$ (which implies that dots and accents have to be handled separately), the set $\mathcal{S}$ can be limited to a set of sequences of contiguous regions without any loss in performance.

Zimmerman and Matas [9] showed that individual characters are often Category Specific Extremal Regions (CSERs) and Donoser et al. [10] further demonstrated that characters are detected as Maximally Stable Extremal Regions (MSERs) [8]. In [6], we show that detection rate of MSERs is improved if multiple projections are used.

In this paper, we extend this approach by using whole tree of MSER lattice induced by the inclusion relation, in contrast to [6] where only root nodes (i.e. supremums of the MSER lattice) were considered which implied that a high MSER margin had to be used to maximize the number of root nodes which correspond to letters. If a lower margin is used, the MSER detector finds more regions but only certain regions correspond to characters. As shown in Figure 1, smaller MSERs are embedded into bigger ones, thus forming a tree where only certain combinations of nodes can be selected as letters, because in a word one letter cannot be embedded into another. We refer to individual nodes of the MSER tree as MSER++ to emphasize that multiple projections (gray, red, green and blue channel) are used and the internal tree structure is taken into account.

C. Exhaustive search

Let $\mathcal{M}$ denote the set of MSER++ in the image $\mathbf{I}$. Even though the cardinality of $\mathcal{M}$ is linear in number of pixels, the cardinality of the set $\mathcal{S}$ of all sequences is still exponential (the complexity has decreased from 2^{2^n} to 2^n only).

Let $\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_n$ denote "upper-bound" verification functions which determine whether s^L is a subsequence of a text sequence or a text sequence itself

$$\hat{v}_L(s^L) = 1 \iff \exists s' : s^L \subseteq s',\ v(s') = 1 \qquad (1)$$

It follows that the enumeration of $\mathcal{E} = \{w \in \mathcal{S} : v(w) = 1\}$ can be equivalently defined as finding the set of unextendable sequences

$$\hat{\mathcal{E}} = \bigcup_{L=1}^{n} \{w \in \mathcal{E}^L :\ \forall s' \supset w \in \mathcal{E}^{L+1}\ \hat{v}_{L+1}(s') = 0\} \qquad (2)$$

where $\mathcal{E}^1, \mathcal{E}^2, \ldots, \mathcal{E}^n$ denote sets of text (sub)sequences of length L

$$\begin{aligned}
\mathcal{E}^1 &= \{r \in \mathcal{M} \mid \hat{v}_1(r) = 1\} \\
\mathcal{E}^2 &= \{(r_1, r_2) \mid r_1, r_2 \in \mathcal{E}^1, r_1 \neq r_2, \hat{v}_2(r_1, r_2) = 1\} \\
\mathcal{E}^3 &= \{(r_1, r_2, r_3) \mid (r_1, r_2), (r_2, r_3) \in \mathcal{E}^2, \\
&\quad r_i \neq r_j\ \forall i, j, \hat{v}_3(r_1, r_2, r_3) = 1\} \quad \ldots \\
\mathcal{E}^n &= \{(r_1, r_2, \ldots, r_n) \mid (r_1, r_2, \ldots, r_{n-1}), \\
&\quad (r_2, r_3, \ldots, r_n) \in \mathcal{E}^{n-1}, r_i \neq r_j\ \forall i, j, \hat{v}_n(r_1, r_2, \ldots, r_n) = 1\}
\end{aligned} \qquad (3)$$

This decomposition allows efficient pruning of the exhaustive search, because non-text subsequences are excluded without a need to build a complete sequence, which prevents from a combinatorial explosion of enumerating the $\mathcal{S}^L$ sets of all sequences of length L.

D. Verification functions

The choice of upper-bound verification functions $\hat{v}_L$ is crucial for the proposed method. Since the optimal verification function $v^*(s)$ is not known, the upper-bound verification functions $\hat{v}_L$ have to be approximated. The key criteria for the approximation is achieving high recall while rejecting as many non-text sequences as possible to prune the search and limit the size of the $\mathcal{E}^L$ sets.

The function $\hat{v}_1(r)$ is a SVM character classifier, which determines whether the region is a character or not based on a set of region measurements (height ratio, compactness, etc.) - see [6]. The function is scale invariant, but not rotation invariant so possible rotations had to be included in the training set. On average, the $\hat{v}_1$ function correctly includes 83% of text regions whilst it correctly excludes 93% of non-text regions such as plants, trees or other random textures.

The $\hat{v}_2(r_1, r_2)$ function consists of pairwise rules which compare measurements of the two regions. The rules require that height ratio, centroid angle and region distance normalized by region width fall within a given interval obtained in a training stage (similar binary rules have been used in many previous works [5], [6], [7], [4]).

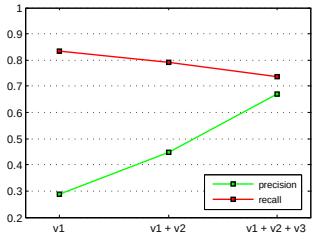


Figure 2. The effects of applying the verification functions $\hat{v}_1$, $\hat{v}_2$ and $\hat{v}_3$ on individual character localization performance

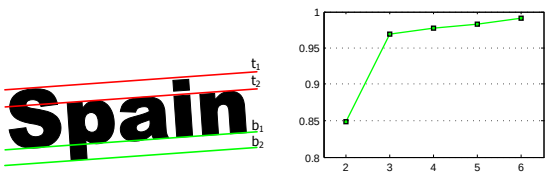
Table II
AVERAGE VERIFICATION FUNCTION CHARACTERISTICS

Function	precision (%)	recall (%)	pruned (%)	total time (s)
$\hat{v}_1$	28.9	83.2	93.3	0.21
$\hat{v}_2$	61.6	94.2	97.0	0.01
$\hat{v}_3$	78.5	87.2	37.2	0.12

In the proposed method, a new rule which exploits the structure of MSER lattice is added. As demonstrated in Figure 1, two regions cannot be in the same sequence if there is a (transitive) parent-child relationship between them, as in this case the first region is embedded into the second one or vice-versa, which is extremely rare for standard text.

In experiments, the $\hat{v}_2$ function pruned out on average 97% of region pairs in an image, but still a significant number of region pairs which are not text passed through (precision is only 62%) - see Table II. This is caused by the fact that the individual measurements on two regions can greatly differ (for instance, the height ratio between the leading capital letter and following lower-case letters can be greater than 3 in some words, the color of two subsequent letters can differ a lot because of lighting conditions, etc.), so a very conservative (i.e. large) interval has to be used to support this variety of texts.

Since the implemented pairwise rules are not sufficiently selective, higher-order features have to be used to reduce the number of false positives. One of such features is based on the observation that letters in a word can be fitted by one or more top and bottom lines (see Figure 3a) and distance of individual letters from these lines is limited (subject to normalization by region height). We refer to this set of lines as *word text lines*.



(a)

(b)

Figure 3. Word text line parameters (a). Dependence of text line parameters estimate accuracy on sequence length (b)

Word text lines estimate $\tau = (t^1, t^2, b^1, b^2, x^{\min}, x^{\max}, h)$ is obtained by inferring a direction k of the text by fitting bottom points using Least-Median of Squares and then fitting top and bottom points of all regions with at most two

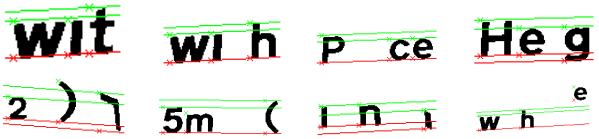


Figure 4. Word text line estimates and triplets accepted (top row) and rejected (bottom row) by the $\hat{v}_3$ function. Top points and lines marked green, bottom points and lines marked red

top (t^1, t^2) and bottom (b^1, b^2) lines with inferred direction k in order to obtain minimal square error (see Figure 4). Variables $x^{\min}$, $x^{\max}$, $h^{\max}$ denote left and right boundary of the word and word height.

In order to obtain a direction of the text, at least 2 regions are needed, but this estimate can be very inaccurate (e.g. fitting bottom points of letters "ly" will result in an incorrect direction, because bottom points of each letter 'sit' on a different bottom line). If three regions are used the estimate is more accurate (see Figure 3b). This fact is exploited by the verification function $\hat{v}_3(r_1, r_2, r_3)$ which creates a word text line estimate τ for given triplet and then verifies that the estimate is valid (mutual vertical distance of the text lines is constrained based on thresholds created during training) and that distance of all three regions from τ is within an interval obtained in a training stage. The recall of after applying the $\hat{v}_3$ function is 87% and 37% of region pairs are pruned out.

The concept of word text lines was used for baseline estimation in printed document analysis [11] and was also applied to text localization in [12]. In the proposed method, only triplets of regions are always used to infer these parameters, in contrast to previous methods where these parameters are estimated on whole words.

As demonstrated in Figure 3b, increasing the number of regions in a sequence does not significantly improve the estimate, which suggests that the geometrical parameters of the word apply to all its subsequences as well. Based on this observation, the verification functions $\hat{v}_4, \ldots, \hat{v}_n$ are approximated by verifying that the text line parameters of all subsequences of length 3 are consistent:

$$\hat{v}_L(s^L) = 1 \iff \forall s_1^3, s_2^3 \subset s^L : d(s_1^3, s_2^3) < \theta \qquad (4)$$

The distance $d(s_1^3, s_2^3)$ of two triplets (see Equation 5) is defined as the largest normalized vertical difference of their text line parameter estimates τ at their boundary points. The function $\hat{v}_L(s^L)$ accepts the sequence s if distance between all triplets in the sequence s is below a threshold θ, which is a parameter of the method obtained during training.

Only the smallest distance is taken into account for top lines as some triplets may contain incomplete set of text lines - for instance in the word "Bear" the triplets "Bea", "Bar" and "Ber" have two top lines because of the capital letter "B", whereas the triplet "ear" has only one top line, which can match to any of the two top lines in the remaining triplets. The same argument applies to bottom lines (e.g. "space") and the two situations can even occur at the same

Figure 5. Exhaustive search for text (sub)sequences. Each sequence is displayed by its region centroids (with random noise to avoid overlapping)

time (e.g. "Gray"), however a bottom line is never matched to a top line or vice-versa.

$$d(s_1^3, s_2^3) = d(\tau_1, \tau_2) = \max\left(\min_{i,j=1\ldots2} \delta(t_1^i, t_2^j), \min_{i,j=1\ldots2} \delta(b_1^i, b_2^j)\right) \quad (5)$$

$$\delta(a,b) = \frac{\max\left(|a(x) - b(x)|, |a(x') - b(x')|\right)}{h}$$
$$x = \min(x_1^{\min}, x_2^{\min}) \quad x' = \max(x_1^{\max}, x_2^{\max})$$
$$h = \max(h_1, h_2) \quad (6)$$

In order to overcome low recall of the $\hat{v}_1$ function, the text localization is performed twice: in the first run, all verification functions are taken into account to build initial text line hypotheses. In the second run, v_1 is forced to 1 and only existing text line hypotheses are taken into account (using the hypotheses-verification framework [6]). The recall of $\hat{v}_2, \ldots, \hat{v}_n$ is not as crucial as one region can be present in multiple subsequences.

The verification function approximation does not guarantee that one region is an element of one sequence only. If this situation occurs, the longer sequence is selected and the other conflicting sequences are discarded. This can be seen as a voting process where each sequence votes for its direction and the most significant direction wins. This process effectively eliminates false positives which are not consistent with text line direction (see Figure 5, bottom-right). Let $\mathcal{E}'$ denote a set of estimates $\mathcal{E}$ without conflicting sequences.

In the proposed method, only sequences longer than 3 regions are accepted because of the low individual precision of the $\hat{v}_1$ and $\hat{v}_2$ functions and the inability to utilize the text line geometric features with individual characters or character pairs.

Table III
TEXT LOCALIZATION (TOP) AND RECOGNITION (BOTTOM) RESULTS ON THE ICDAR 2003 DATASET

method	precision	recall	f
proposed method	**0.65**	**0.64**	**0.63**
Hinnerk Becker [14]	0.62	0.67	0.62
Alex Chen [14]	0.60	0.60	0.58
Neumann and Matas [6]	0.59	0.55	0.57
proposed method	**0.72***	**0.62***	**0.67***
Epshtein et al. [5]	0.73*	0.60*	0.66*
Pan et al. [4]	(0.71)	(0.67)	N/A
Zhang et al. [7]	(0.73)	(0.62)	N/A

method	precision	recall	f
proposed method	**0.42**	**0.41**	**0.41**
Neumann and Matas [6]	0.42	0.39	0.40

III. EXPERIMENTS

The proposed method was evaluated on the most cited ICDAR 2003 dataset [13], which contains 249 images with text of varying sizes and positions.

The standard evaluation protocol defined in [13] was used. The protocol uses words as the unit for comparison, where bounding boxes of words output by the evaluated method $\mathcal{E}$ (*estimates*) are compared to the ground truth T (*targets*). The protocol uses the notion of a flexible match of a region r in a set of regions $\mathcal{R}$ as $m(r, \mathcal{R}) = \max_{r' \in \mathcal{R}} m_p(r, r')$, where $m_p(r, r')$ denotes the area of intersection divided by the area of the minimum bounding box containing both rectangles. Precision and recall of text localization are defined as

$$p_l = \frac{\sum_{r_e \in \mathcal{E}} m(r_e, \mathcal{T})}{|\mathcal{E}|} \quad r_l = \frac{\sum_{r_t \in \mathcal{T}} m(r_t, \mathcal{E})}{|\mathcal{T}|} \quad (7)$$

and a standard f-measure is used to combine both figures.

All performance measures are calculated on each image independently and then an average value over all images is taken as performance of the method. The proposed method achieves precision of 0.65 and recall of 0.64, which outperforms the existing methods as shown in Table III.

This performance measure was used in the ICDAR 2003 and 2005 competitions [13], [14], however papers presented later deviate from the original performance measure. In [5], only one precision and recall value over the whole set of estimates and targets is calculated (marked with an asterisk in Table III for comparison), which gives higher weight to images with more words, which typically leads to better results on the ICDAR dataset as the more challenging images in the dataset contain only small number of words. Other papers [7], [4] use whole text lines for evaluation, so direct comparison is not possible (results given in parentheses in Table III).

The localization output of the proposed method was passed to recognition modules of the hypotheses-verification

Figure 6. Text localization and recognition examples on the ICDAR 2003 dataset. The performance measure has the format $(p_l, r_l; p_r, r_r)$. Notice the low number of false positives despite textures in the images and robustness against blur and reflections. Incorrectly recognized letters marked red

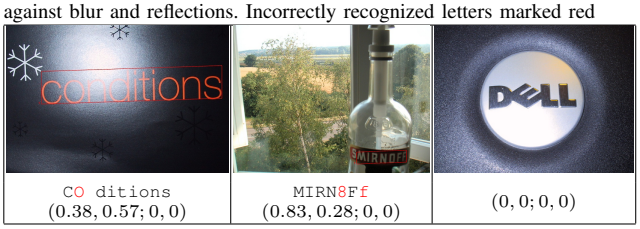


Figure 7. Limitations of the proposed method. Reflection of a flash is too strong so the letter "n" is not detected as an MSER (left). An unsupported text line shape and letters written on glass not detected as MSERs (middle). Multiple letters joint into one region (right)

framework [6]. Table III shows text recognition precision (p_r) and recall (r_r), which are only slightly improved over the previous method, because the text recognition evaluation uses very strict metric, so even a significant improvement in text localization does not guarantee that significantly more words will be recognized without any mistake.

IV. Conclusions

An efficient method for text localization in real-word images was introduced. It was demonstrated that suitable selection of verification functions that control the grouping allows exhaustive search of the space of all character sequences to such an extent that the text can be localized and recognized in real time. The method exploits higher-order features, which significantly improves its performance and accuracy.

Additionally, the method includes a novel selector of MSERs which thanks to exploiting region topology allows using lower margin for detection, which improved individual character detection rate from 84.0% to 95.7% without any impact on calculation complexity.

On the highly cited ICDAR dataset [13], the method achieved precision of 0.65 and recall of 0.64 which represents state-of-the-art results in text localization. The precision 0.42 and recall 0.41 of text recognition is also better than our previous method, however the improvement is only marginal as recognition modules are identical to [6]. On a standard PC, the text localization and recognition took on average 830ms per image on the ICDAR dataset (200ms on average for 640×480 images).

Acknowledgment

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

References

[1] X. Lin, "Reliable OCR solution for digital content remastering," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Dec. 2001.

[2] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," *CVPR*, vol. 2, pp. 366–373, 2004.

[3] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 256 –268, 2002.

[4] Y.-F. Pan, X. Hou, and C.-L. Liu, "Text localization in natural scene images based on conditional random field," in *ICDAR 2009*. IEEE Computer Society, 2009, pp. 6–10.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *CVPR 2010*, pp. 2963 –2970.

[6] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *ACCV 2010*, ser. LNCS 6495, vol. IV, November 2010, pp. 2067–2078.

[7] J. Zhang and R. Kasturi, "Character energy and link energy-based text extraction in scene images," in *ACCV 2010*, ser. LNCS 6495, vol. II, November 2010, pp. 832–844.

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, pp. 761–767, 2004.

[9] J. Matas and K. Zimmermann, "A new class of learnable detectors for categorisation," in *Image Analysis*, ser. LNCS, 2005, vol. 3540, pp. 541–550.

[10] M. Donoser, H. Bischof, and S. Wagner, "Using web search engines to improve text recognition," in *ICPR 2008*, pp. 1 –4.

[11] T. Caesar, J. Gloger, and E. Mandler, "Estimating the baseline for written material," in *ICDAR 1995*, vol. 1, pp. 382–385.

[12] L. Neumann and J. Matas, "Estimating hidden parameters for text localization and recognition," in *Proc. of 16th CVWW*, February 2011, pp. 29–26.

[13] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *ICDAR 2003*, 2003, p. 682.

[14] S. M. Lucas, "Text locating competition results," *Document Analysis and Recognition, International Conference on*, vol. 0, pp. 80–85, 2005.

D Real-Time Scene Text Localization and Recognition

A reprint of our article *Real-Time Scene Text Localization and Recognition* presented at the 25th Conference on Computer Vision and Pattern Recognition (CVPR 2012).

Real-Time Scene Text Localization and Recognition

Lukáš Neumann Jiří Matas
Centre for Machine Perception, Department of Cybernetics
Czech Technical University, Prague, Czech Republic
neumalu1@cmp.felk.cvut.cz, matas@cmp.felk.cvut.cz

Abstract

An end-to-end real-time scene text localization and recognition method is presented. The real-time performance is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). The ER detector is robust to blur, illumination, color and texture variation and handles low-contrast text.

In the first classification stage, the probability of each ER being a character is estimated using novel features calculated with $O(1)$ complexity per region tested. Only ERs with locally maximal probability are selected for the second stage, where the classification is improved using more computationally expensive features. A highly efficient exhaustive search with feedback loops is then applied to group ERs into words and to select the most probable character segmentation. Finally, text is recognized in an OCR stage trained using synthetic fonts.

The method was evaluated on two public datasets. On the ICDAR 2011 dataset, the method achieves state-of-the-art text localization results amongst published methods and it is the first one to report results for end-to-end text recognition. On the more challenging Street View Text dataset, the method achieves state-of-the-art recall. The robustness of the proposed method against noise and low contrast of characters is demonstrated by "false positives" caused by detected watermark text in the dataset.

1. Introduction

Text localization and recognition in real-world (scene) images is an open problem which has been receiving significant attention since it is a critical component in a number of computer vision applications like searching images by their textual content, reading labels on businesses in map applications (e.g. Google Street View) or assisting visually impaired. Several contests have been held in the past years [10, 9, 20] and the winning method in the most recent ICDAR 2011 contest was able to localize only 62% words correctly [20] despite the fact that the dataset is not fully realistic (words are horizontal only, they occupy a significant part of the image, there is no perspective distortion or significant noise).

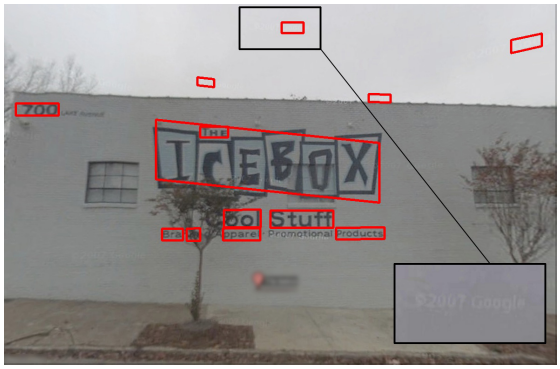


Figure 1. Text detection in the SVT dataset. All "false positives" in the image are caused by watermarks embedded into the dataset. This demonstrates robustness of the proposed method against noise and low contrast of characters (in the bottom-right corner the area of interest is enlarged and contrast artificially increased, "©2007 Google" is readable).

Localizing text in an image is potentially a computationally very expensive task as generally any of the 2^N subsets can correspond to text (where N is the number of pixels). Text localization methods deal with this problem in two different ways.

Methods based on a sliding window [6, 2, 7] limit the search to a subset of image rectangles. This reduces the number of subsets checked for the presence of text to cN where c is a constant that varies between very small values (< 1) for single-scale single-rotation methods to relatively large values ($\gg 1$) for methods that handle text with a different scale, aspect, rotation, skew, etc.

Methods in the second group [5, 17, 14, 15, 24] find individual characters by grouping pixels into regions using connected component analysis assuming that pixels belonging to the same character have similar properties. Connected component methods differ in the properties used (color, stroke-width, etc.). The advantage of the connected component methods is that their complexity typically does not depend on the properties of the text (range of scales, orientations, fonts) and that they also provide a segmentation

which can be exploited in the OCR step. Their disadvantage is a sensitivity to clutter and occlusions that change connected component structure.

In this paper, we present an end-to-end real-time[1] text localization and recognition method which achieves state-of-the-art results on standard datasets. The real-time performance is achieved by posing the character detection problem as an efficient sequential selection from the set of Extremal Regions (ERs). The ER detector is robust against blur, low contrast and illumination, color and texture variation[2]. Its complexity is $O(2pN)$, where p denotes number of channels (projections) used.

In the first stage of the classification, the probability of each ER being a character is estimated using novel features calculated with $O(1)$ complexity and only ERs with locally maximal probability are selected for the second stage, where the classification is improved using more computationally expensive features. A highly efficient exhaustive search with feedback loops (adapted from [15]) is then applied to group ERs into words and select the most probable character segmentation.

Additionally, a novel gradient magnitude projection which allows edges to induce ERs is introduced. It is further demonstrated that by inclusion of the gradient projection 94.8% of characters are detected by the ER detector.

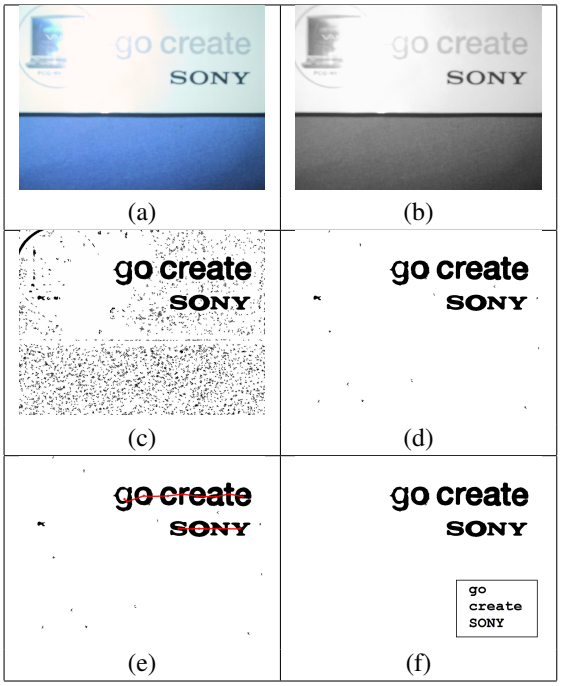
The rest of the document is structured as follows: In Section 2, an overview of previously published methods is given. Section 3 describes the proposed method. In Section 4, the experimental evaluation is presented. The paper is concluded in Section 5.

2. Previous Work

Numerous methods which focus solely on text localization in real-world images have been published [6, 2, 7, 17]. The method of Epstein et al. in [5] converts an input image to a greyscale space and uses Canny detector [1] to find edges. Pairs of parallel edges are then used to calculate stroke width for each pixel and pixels with similar stroke width are grouped together into characters. The method is sensitive to noise and blurry images because it is dependent on a successful edge detection and it provides only single segmentation for each character which not necessarily might be the best one for an OCR module. A similar edge-based approach with different connected component algorithm is presented in [24]. A good overview of the methods and their performance can be also found in ICDAR Robust Reading competition results [10, 9, 20].

Only a few methods that perform both text localization and recognition have been published. The method of Wang et al. [21] finds individual characters as visual words using the sliding-window approach and then uses a lexicon to group characters into words. The method is able to cope with noisy data, but its generality is limited as a lexicon of words (which contains at most 500 words in their experiments) has to be supplied for each individual image.

Methods presented in [14, 15] detect characters as Maximally Stable Extremal Regions (MSERs) [11] and perform text recognition using the segmentation obtained by the MSER detector. An MSER is an particular case of an Extremal Region whose size remains virtually unchanged over a range of thresholds. The methods perform well but have problems on blurry images or characters with low contrast. According to the description provided by the ICDAR 2011 Robust Reading competition organizers [20] the winning method is based on MSER detection, but the method



	run time(ms)	No. of ERs
Initial image	-	6×10^6
Classification (1^{st} stage)	1120	2671
Classification (2^{nd} stage)	130	20
Region grouping	20	12
OCR	110	12

(g)

Figure 2. Text localization and recognition overview. (a) Source 2MPx image. (b) Intensity channel extracted. (c) ERs selected in $O(N)$ by the first stage of the sequential classifier. (d) ERs selected by the second stage of the classifier. (e) Text lines found by region grouping. (f) Only ERs in text lines selected and text recognized by an OCR module. (g) Number of ERs at the end of each stage and its duration

[1] We consider a text recognition method real-time if the processing time is comparable with the time it would take a human to read the text.

[2] A www service allowing testing of the method is available at http://cmp.felk.cvut.cz/ neumalu1/TextSpotter

(a) (b) (c)

Figure 3. Intensity gradient magnitude channel ∇. (a) Source image. (b) Projection output. (c) Extremal Regions at threshold $\theta = 24$ (ERs bigger than 30% of the image area excluded for better visualization)

itself had not been not published and it does not perform text recognition.

The proposed method differs from the MSER-based methods [14, 15] in that it tests all ERs (not only the subset of MSERs) while reducing the memory footprint and maintaining the same computational complexity and real-time performance. The idea of dropping the stability requirement of MSERs and selecting a class-specific (not necessarily stable) Extremal Regions was first presented by Zimmermann and Matas [12], who used image moments as features for a monolithic neural network, which was trained for a given set of shapes (e.g. textures, specific characters). In our method, the selection of suitable ERs is carried out in real-time by a sequential classifier on the basis of novel features which are specific for character detection. Moreover, the classifier is trained to output probability and thus extracts several segmentations of a character.

3. The Proposed Method

3.1. Extremal Regions

Let us consider an image $\mathbf{I}$ as a mapping $\mathbf{I} : \mathcal{D} \subset \mathbb{N}^2 \rightarrow \mathcal{V}$, where $\mathcal{V}$ typically is $\{0, \ldots, 255\}^3$ (a color image). A channel $\mathbf{C}$ of the image $\mathbf{I}$ is a mapping $\mathbf{C} : \mathcal{D} \rightarrow \mathcal{S}$ where $\mathcal{S}$ is a totally ordered set and $f_c : \mathcal{V} \rightarrow \mathcal{S}$ is a *projection* of pixel values to a totally ordered set. Let A denote an adjacency (neighborhood) relation $A \subset \mathcal{D} \times \mathcal{D}$. In this paper we consider 4-connected pixels, i.e. pixels with coordinates $(x \pm 1, y)$ and $(x, y \pm 1)$ are adjacent to the pixel (x, y).

Region $\mathcal{R}$ of an image $\mathbf{I}$ (or a channel $\mathbf{C}$) is a contiguous subset of $\mathcal{D}$, i.e. $\forall p_i, p_j \in \mathcal{R}\ \exists p_i, q_1, q_2, \ldots, q_n, p_j : p_i A q_1, q_1 A q_2, \ldots, q_n A p_j$. Outer region boundary $\partial\mathcal{R}$ is a set of pixels adjacent but not belonging to $\mathcal{R}$, i.e. $\partial\mathcal{R} = \{p \in \mathcal{D} \setminus \mathcal{R} : \exists q \in \mathcal{R} : pAq\}$. *Extremal Region* (ER) is a region whose outer boundary pixels have strictly higher values than the region itself, i.e. $\forall p \in \mathcal{R}, q \in \partial\mathcal{R} : \mathbf{C}(q) > \theta \geq \mathbf{C}(p)$, where θ denotes threshold of the Extremal Region.

An ER r at threshold θ is formed as a union of one or more (or none) ERs at threshold $\theta - 1$ and pixels of value θ, i.e. $r = \left(\bigcup u \in R_{\theta-1}\right) \cup \left(\bigcup p \in \mathcal{D} : \mathbf{C}(p) = \theta\right)$, where $R_{\theta-1}$ denotes set of ERs at threshold $\theta - 1$. This induces an *inclusion relation* amongst ERs where a single ER has one or more predecessor ERs (or no predecessor if it contains only pixels of a single value) and exactly one successor ER (the ultimate successor is the ER at threshold 255 which contains all pixels in the image).

Channel	R (%)	P (%)
R	83.3	7.7
G	85.7	10.3
B	85.5	8.9
H	62.0	2.0
S	70.5	4.1
I	85.6	10.1
∇	74.6	6.3

Channel	R (%)	P (%)
I∪H	89.9	6.0
I∪S	90.1	7.2
I∪∇	90.8	8.4
I∪H∪S	92.3	5.5
I∪H∪∇	93.1	6.1
I∪R∪G∪B	90.3	9.2
I∪H∪S∪∇	**93.7**	**5.7**
all (7 ch.)	94.8	7.1

Table 1. Recall (R) and precision (R) of character detection by ER detectors in individual channels and their combinations. The channel combination used in the experiments is in bold

In this paper, we consider RGB and HSI color spaces [3] and additionally an *intensity gradient* channel (∇) where each pixel is assigned the value of "gradient" approximated by maximal intensity difference between the pixel and its neighbors (see Figure 3):

$$\mathbf{C}_\nabla(p) = \max_{q \in \mathcal{D} : pAq} \left\{ |\mathbf{C}_\mathbf{I}(p) - \mathbf{C}_\mathbf{I}(q)| \right\}$$

An experimental validation shows that up to 85.6% characters are detected as ERs in a single channel and that 94.8% characters are detected if the detection results are combined from all channels (see Table 1). A character is considered as detected if bounding box of the ER matches at least 90% of the area of the bounding box in the ground truth. In the proposed method, the combination of intensity (`I`), intensity gradient (∇), hue (`H`) and saturation (`S`) channels was used as it was experimentally found as the best trade-off between short run time and localization performance.

3.2. Incrementally Computable Descriptors

The key prerequisite for fast classification of ERs is a fast computation of region descriptors that serve as features for the classifier. As proposed by Zimmerman and Matas [12], it is possible to use a particular class of descriptors and exploit the inclusion relation between ERs to incrementally compute descriptor values.

Let $R_{\theta-1}$ denote a set of ERs at threshold $\theta - 1$. An ER $r \in R_\theta$ at threshold θ is formed as a union of pixels of regions at threshold $\theta - 1$ and pixels of value θ, i.e. $r = \left(\bigcup u \in R_{\theta-1}\right) \cup \left(\bigcup p \in \mathcal{D} : \mathbf{C}(p) = \theta\right)$. Let us further assume that descriptors $\phi(u)$ of all ERs at threshold $u \in R_{\theta-1}$ are already known. In order to compute a descriptor $\phi(r)$ of the region $r \in R_\theta$ it is necessary to combine descriptors of regions $u \in R_{\theta-1}$ and pixels $\{p \in \mathcal{D} : \mathbf{C}(p) = \theta\}$ that formed the region r, i.e.

$\phi(r) = \Big(\oplus\phi(u)\Big) \oplus \Big(\oplus\psi(p)\Big)$, where $\oplus$ denotes an operation that combines descriptors of the regions (pixels) and $\psi(p)$ denotes an initialization function that computes the descriptor for given pixel p. We refer to such descriptors where $\psi(p)$ and $\oplus$ exist as *incrementally computable* (see Figure 4).

It is apparent that one can compute descriptors of all ERs simply by sequentially increasing threshold θ from 0 to 255, calculating descriptors ψ for pixels added at threshold θ and reusing the descriptors of regions ϕ at threshold $\theta - 1$. Note that the property implies that it is necessary to only keep descriptors from the previous threshold in the memory and that the ER method has a significantly smaller memory footprint when compared with MSER-based approaches. Moreover if it is assumed that the descriptor computation for a single pixel $\psi(p)$ and the combining operation $\oplus$ has constant time complexity, the resulting complexity of computing descriptors of all ERs in an image of N pixels is $O(N)$, because $\phi(p)$ is computed for each pixel just once and combining function can be evaluated at most N times, because the number of ERs is bounded by the number of pixels in the image.

In this paper we used the following incrementally computed descriptors:

Area a. Area (i.e. number of pixels) of a region. The initialization function is a constant function $\psi(p) = 1$ and the combining operation $\oplus$ is an addition $(+)$.

Bounding box $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$. Top-right and bottom-left corner of the region. The initialization function of a pixel p with coordinates (x, y) is a quadruple $(x, y, x + 1, y + 1)$ and the combining operation $\oplus$ is $(\min, \min, \max, \max)$ where each operation is applied to its respective item in the quadruple. The width w and height h of the region is calculated as $x_{\max} - x_{\min}$ and $y_{\max} - y_{\min}$ respectively.

Perimeter p. The length of the boundary of the region (see Figure 4a). The initialization function $\psi(p)$ determines a change of the perimeter length by the pixel p at the threshold where it is added, i.e. $\psi(p) = 4 - 2|\{q : qAp \wedge \mathbf{C}(q) \leq \mathbf{C}(p)\}|$ and the combining operation $\oplus$ is an addition $(+)$. The complexity of $\psi(p)$ is $O(1)$, because each pixel has at most 4 neighbors.

Euler number η. Euler number (genus) is a topological feature of a binary image which is the difference between the number of connected components and the number of holes. A very efficient yet simple algorithm [18] calculates the Euler number by counting 2×2 pixel patterns called quads. Consider the following patterns of a binary image:

$$Q_1 = \left\{\begin{matrix}1 & 0\end{matrix}, \begin{matrix}0 & 0\end{matrix}, \begin{matrix}0 & 0\end{matrix}, \begin{matrix}1 & 0\end{matrix}, \begin{matrix}0 & 1\end{matrix}, \begin{matrix}0 & 1\end{matrix}, \begin{matrix}0 & 1\end{matrix}, \begin{matrix}0 & 0\end{matrix}\right\}$$

$$Q_2 = \left\{\begin{matrix}0 & 1\end{matrix}, \begin{matrix}1 & 1\end{matrix}, \begin{matrix}1 & 1\end{matrix}, \begin{matrix}0 & 1\end{matrix}, \begin{matrix}1 & 1\end{matrix}, \begin{matrix}1 & 0\end{matrix}, \begin{matrix}1 & 0\end{matrix}, \begin{matrix}1 & 1\end{matrix}\right\}$$

$$Q_3 = \left\{\begin{matrix}0 & 1\end{matrix}, \begin{matrix}1 & 0\end{matrix}, \begin{matrix}1 & 0\end{matrix}, \begin{matrix}0 & 1\end{matrix}\right\}$$

Euler number is then calculated as

$$\eta = \frac{1}{4}\left(C_1 - C_2 + 2C_3\right))$$

where C_1, C_2 and C_3 denote number of quads Q_1, Q_2 and Q_3 respectively in the image.

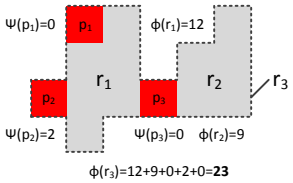
It follows that the algorithm can be exploited for incremental computation by simply counting the change in the number of quads in the image. The value of the initialization function $\psi(p)$ is determined by the change in the number of the quads Q_1, Q_2 and Q_3 by changing the value of the pixel p from 0 to 1 at given threshold $\mathbf{C}(p)$ (see Figure 4b), i.e. $\psi(p) = \frac{1}{4}\left(\Delta C_1 - \Delta C_2 + 2\Delta C_3\right))$. The complexity of $\psi(p)$ is $O(1)$, because each pixel is present in at most 4 quads. The combining operation $\oplus$ is an addition $(+)$.

Horizontal crossings $\mathbf{c}_i$. A vector (of length h) with number of transitions between pixels belonging ($p \in r$) and not belonging ($p \notin r$) to the region in given row i of the region r (see Figure 4c and 7). The value of the initialization function is given by the presence/absence of left and right neighboring pixels of the pixel p at the threshold $\mathbf{C}(p)$. The combining operation $\oplus$ is an element-wise addition $(+)$ which aligns the vectors so that the elements correspond to same rows. The computation complexity of $\psi(p)$ is constant (each pixel has at most 2 neighbors in the horizontal direction) and the element-wise addition has constant complexity as well assuming that a data structure with $O(1)$ random access and insertion at both ends (e.g. double-ended queue in a growing array) is used.

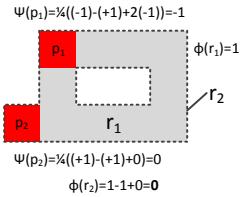
3.3. Sequential Classifier

In the proposed method, each channel is iterated separately (in the original and inverted projections) and subsequently ERs are detected. In order to reduce the high false positive rate and the high redundancy of the ER detector, only distinctive ERs which correspond to characters are selected by a sequential classifier. The classification is broken down into two stages for better computational efficiency (see Figure 2).

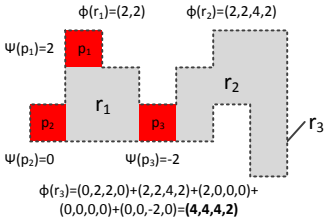
In the first stage, a threshold is increased step by step from 0 to 255, incrementally computable descriptors (see Section 3.2) are computed (in $O(1)$) for each ER r and the descriptors are used as features for a classifier which estimates the class-conditional probability $p(r|\text{character})$. The value of $p(r|\text{character})$ is tracked using the inclusion relation of ER across all thresholds (see Figure 6) and only the ERs which correspond to local maximum of the probability $p(r|\text{character})$ are selected (if the local maximum of the probability is above a global limit $p_{\min}$ and the difference between local maximum and local minimum is greater than $\Delta_{\min}$).



(a) perimeter p



(b) Euler number η



(c) Horizontal crossings $\mathbf{c}_i$

Figure 4. Incrementally computable descriptors. Regions already existing at threshold $\theta - 1$ marked grey, new pixels at threshold θ marked red, the resulting region at threshold θ outlined with a dashed line

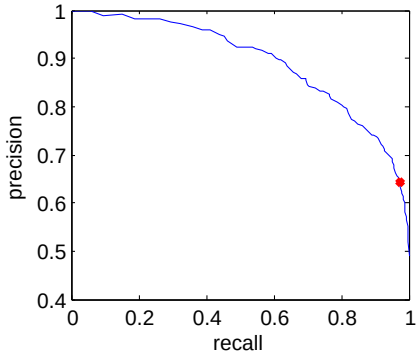
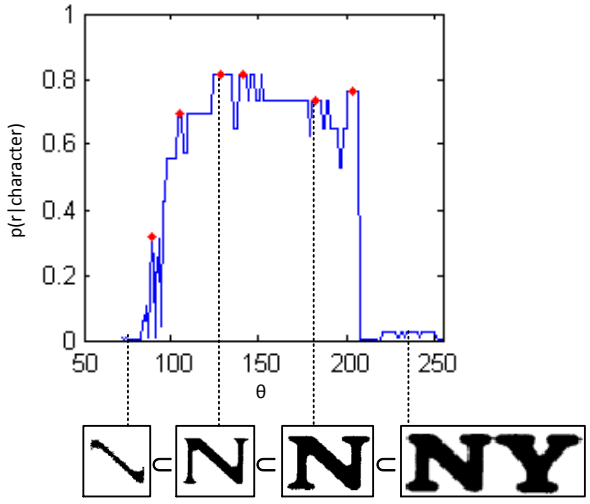


Figure 5. The ROC curve of the first stage of the sequential classifier obtained by cross-validation. The configuration used in the experiments marked red (recall 95.6%, precision 67.3)

In this paper, a Real AdaBoost [19] classifier with decision trees was used with the following features (calculated in $O(1)$ from incrementally computed descriptors): aspect ratio (w/h), compactness ($\sqrt{a}/p$), number of holes ($1 - \eta$) and a horizontal crossings feature ($\hat{c}$ = median $\{\mathbf{c}_{\frac{1}{6}w}, \mathbf{c}_{\frac{3}{6}w}, \mathbf{c}_{\frac{5}{6}w}\}$) which estimates number of character strokes in horizontal projection - see Figure 7. Only a fixed-size subset of $\mathbf{c}$ is sampled so that the computation has a constant complexity. The output of the classifier is calibrated to a probability function $p(r|\text{character})$ using Logistic Correction [16]. The parameters were set experimentally to $p_{\min} = 0.2$ and $\Delta_{\min} = 0.1$ to obtain a high value of recall (95.6%) (see Figure 5).

(a)



(b)

Figure 6. In the first stage of the sequential classification the probability $p(r|\text{character})$ of each ER is estimated using incrementally computable descriptors that exploit the inclusion relation of ERs. (a) A source image cut-out and the initial seed of the ER inclusion sequence (marked with a red cross). (b) The value of $p(r|\text{character})$ in the inclusion sequence, ERs passed to the second stage marked red

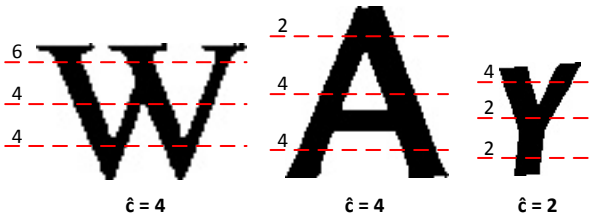


Figure 7. The horizontal crossings feature that is used in the 1st stage of ER classification

In the second stage, the ERs that passed the first stage are classified into character and non-character classes using more informative but also more computationally expensive features. In this paper, an SVM [4] classifier with the RBF kernel [13] was used. The classifier uses all the features calculated in the first stage and the following additional features:

- **Hole area ratio**. a_h/a where a_h denotes number of pixels of region holes. This feature is more informative

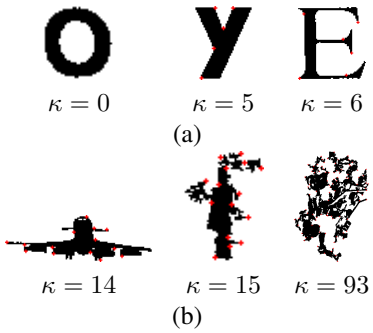


Figure 8. The number of boundary inflexion points κ. (a) Characters. (b) Non-textual content

than just the number of holes (used in the first stage) as small holes in a much larger region have lower significance than large holes in a region of comparable size.

- **Convex hull ratio**. a_c/a where a_c denotes the area of the convex hull of the region.

- **The number of outer boundary inflexion points** κ. The number of changes between concave and convex angle between pixels around the region border (see Figure 8). A character typically has only a limited number of inflexion points ($\kappa < 10$), whereas regions that correspond to non-textual content such as grass or pictograms have boundary with many spikes and thus more inflexion points.

Let us note that all features are scale-invariant, but not rotation-invariant which is why characters of different rotations had to be included in the training set.

3.4. Exhaustive Search

The detector was incorporated into a system described in Neumann and Matas [15], which uses efficiently pruned search to exhaustively search the space of all character sequences in real-time. It exploits higher-order properties of text such as word text lines and its robust grouping stage is able to compensate errors of the character detector. The system was chosen because it is able to handle multiple channels, multiple segmentations for each character (see Figure 6) and to combine detection results from multiple channels using the OCR stage. It also provides text recognition for characters segmented by the character detector. For more details see [15].

4. Experiments

The method was trained using approximately 900 examples of character ERs and 1400 examples of non-character ERs manually extracted the from the ICDAR 2003 training dataset [10] (sequential classifier training) and synthetically generated fonts (OCR stage training). The method was then evaluated with the same parameters on two independent datasets.

method	recall	precision	f
Kim's Method *	62.5	83.0	71.3
proposed method	**64.7**	**73.1**	**68.7**
Yi's Method [23]	58.1	67.2	62.3
TH-TextLoc System [8]	57.7	67.0	62.0
Neumann and Matas [15]	52.5	68.9	59.6

Table 2. Text localization results on the ICDAR 2011 dataset. Unpublished methods marked with a star

4.1. ICDAR 2011 Dataset

The ICDAR 2011 Robust Reading competition dataset [20] contains 1189 words and 6393 letters in 255 images. Using the ICDAR 2011 competition evaluation scheme [22], the method achieves the recall of 64.7%, precision of 73.1% and the f-measure of 68.7% in text localization (see Figure 9 for sample outputs).

The method achieves significantly better recall (65%) than the winner of ICDAR 2011 Robust Reading competition (62%), but the precision (73%) is worse than the winner (83%) and thus the resulting combined f-measure (69%) is worse than the ICDAR 2011 winner (71%), which had not been published. The proposed method however significantly outperforms the second best (published) method of Yi [23] in all three measures (see Table 2). Let us further note that the ICDAR 2011 competition was held in an open mode where authors supply only outputs of their methods on a previously published competition dataset.

Word text recognition recall is 37.2%, precision 37.1% and f-measure 36.5% respectively; a word is considered correctly recognized if it is localized with recall at least 80% and all letters are recognized correctly (case-sensitive comparison) [10]. The average processing time (including text localization) is 1.8s per image on a standard PC.

The word recognition results cannot be compared to any existing method because end-to-end text localization and recognition was not part of the ICDAR 2011 Robust Reading competition and no other method had presented its text recognition results on the dataset.

4.2. Street View Text Dataset

The Street View Text (SVT) dataset [21] contains 647 words and 3796 letters in 249 images harvested from Google Street View. The dataset is more challenging because text is present in different orientations, the variety of fonts is bigger and the images are noisy. The format of the ground truth is different from the previous experiment as only some words are annotated (see Figure 11). Of the annotated words the proposed method achieved a recall of 32.9% using the same evaluation protocol as in the previous section (see Figure 12 for output examples).

The precision of the text localization (19.1%) cannot be

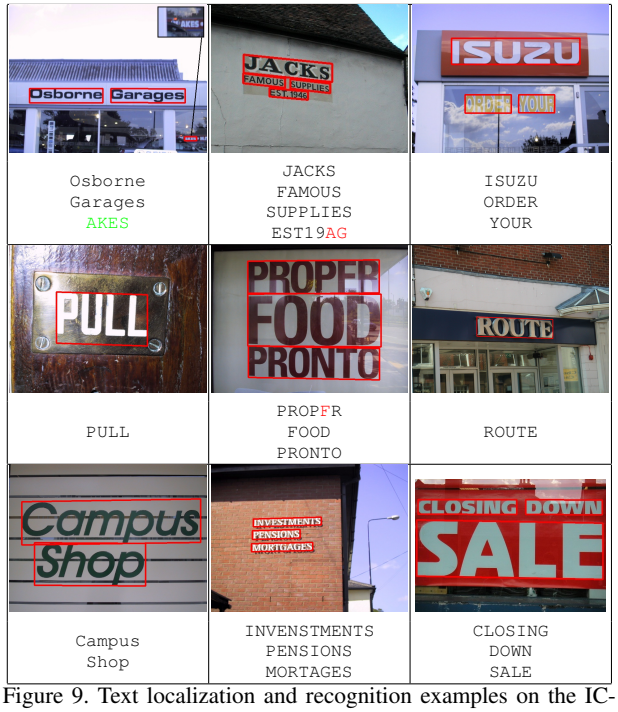


Figure 9. Text localization and recognition examples on the ICDAR 2011 dataset. Notice the robustness against reflections and lines passing through the text (bottom-left). Incorrectly recognized letters marked red, text recognized by the proposed method but not present in the ground truth marked green

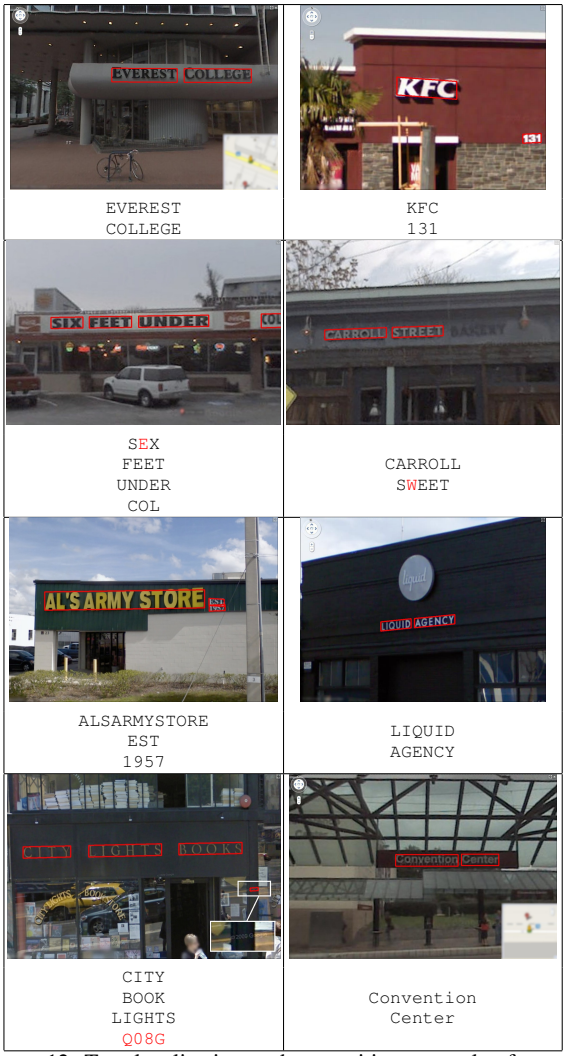
(a) (b) (c)

Figure 10. Problems of the proposed method. (a) Characters with no contrast. (b) Multiple characters joined together. (c) A single letter (the claim that the McDonald's logo is a letter "M" as defined by the annotation is questionable)

taken into account because of the incomplete annotations. It can be observed that many of the false detections are caused by watermark text embedded in each image (see Figure 1), which demonstrates robustness of the proposed method against noise and low contrast of characters.

The results can be compared only indirectly with the method of Wang et al. [21] which using a different evaluation protocol reports the f-measure of 41.0% (achieved for recall 29.0% and precision 67.0%) on the dataset. Moreover the task formulation of the method of Wang et al. differs significantly in that for each image it is given a lexicon of words that should be localized in the image (if present) whereas the proposed method has no prior knowledge about the content of the image and its output is not limited by a fixed lexicon.

Figure 11. Missing annotations in the SVT dataset (annotations marked green, output of the proposed method marked red).



EVEREST COLLEGE | KFC 131

SEX FEET UNDER COL | CARROLL SWEET

ALSARMYSTORE EST 1957 | LIQUID AGENCY

CITY BOOK LIGHTS Q08G | Convention Center

Figure 12. Text localization and recognition examples from the SVT dataset. Notice the high level of noise and the blur (zoomed-in PDF viewing highly recommended). Incorrectly recognized letters marked red.

5. Conclusions

An end-to-end real-time text localization and recognition method is presented in the paper. In the first stage of the classification, the probability of each ER being a character is estimated using novel features calculated with $O(1)$ complexity and only ERs with locally maximal probability are selected for the second stage, where the classification is improved using more computationally expensive features. It is demonstrated that including the novel gradient magnitude projection ERs cover 94.8% of characters. The average run time of the method on a 800×600 image is 0.3s on a standard PC.

The method was evaluated on two public datasets. On the ICDAR 2011 dataset the method achieves state-of-the-art text localization results amongst published methods (recall 64.7%, precision 73.1%, f-measure 68.7%) and we are the first to report results (recall 37.2%, precision 37.1%, f-measure 36.5%) for end-to-end text recognition on the ICDAR 2011 Robust Reading competition dataset.

On the more challenging Street View Text dataset the recall of the text localization (32.9%) can be only compared to the previously published method of Wang et al. [21] (29.0%), however direct comparison is not possible as the method of Wang et al. uses a different task formulation and a different evaluation protocol. Robustness of the proposed method against noise and low contrast of characters is demonstrated by "false positives" caused by detected watermark text in the dataset.

Acknowledgment

The authors were supported by The Czech Science Foundation Project GACR P103/12/G084.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[2] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *CVPR*, 2:366–373, 2004.

[3] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259 – 2281, 2001.

[4] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, March 2000.

[5] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR 2010*, pages 2963 –2970.

[6] L. Jung-Jin, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. In *ICDAR 2011*, pages 429–434, 2011.

[7] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *Circuits and Systems for Video Technology*, 12(4):256 –268, 2002.

[8] H. Liu and X. Ding. Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes. In *ICDAR 2005*, pages 19 – 23 Vol. 1.

[9] S. M. Lucas. Text locating competition results. *Document Analysis and Recognition, International Conference on*, 0:80–85, 2005.

[10] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *ICDAR 2003*, page 682, 2003.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22:761–767, 2004.

[12] J. Matas and K. Zimmermann. A new class of learnable detectors for categorisation. In *Image Analysis*, volume 3540 of *LNCS*, pages 541–550. 2005.

[13] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12:181–201, 2001.

[14] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *ACCV 2010*, volume IV of *LNCS 6495*, pages 2067–2078, November 2010.

[15] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *ICDAR 2011*, pages 687–691, 2011.

[16] A. Niculescu-Mizil and R. Caruana. Obtaining calibrated probabilities from boosting. In *In: Proc. 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[17] Y.-F. Pan, X. Hou, and C.-L. Liu. Text localization in natural scene images based on conditional random field. In *ICDAR 2009*, pages 6–10. IEEE Computer Society, 2009.

[18] W. K. Pratt. *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition, 2001.

[19] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[20] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *ICDAR 2011*, pages 1491–1496, 2011.

[21] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV 2011*, 2011.

[22] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *Int. J. Doc. Anal. Recognit.*, 8:280–296, August 2006.

[23] C. Yi and Y. Tian. Text string detection from natural scenes by structure-based partition and grouping. *Image Processing, IEEE Transactions on*, 20(9):2594 –2605, sept. 2011.

[24] J. Zhang and R. Kasturi. Character energy and link energy-based text extraction in scene images. In *ACCV 2010*, volume II of *LNCS 6495*, pages 832–844, November 2010.

References

[1] R. Beaufort and C. Mancas-Thillou. A weighted finite-state framework for correcting errors in natural scene ocr. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 889 –893, sept. 2007.

[2] A. Bosch, A. Zisserman, and X. Muoz. Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, oct. 2007.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[4] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. *CVPR*, 2:366–373, 2004.

[5] H. Cheng, X. Jiang, Y. Sun, and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34(12):2259 – 2281, 2001.

[6] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. Wu, and A. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 440 – 445, sept. 2011.

[7] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, March 2000.

[8] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. *VISAPP*, 05-08 February 2009, 2009.

[9] M. Donoser, H. Bischof, and S. Wagner. Using web search engines to improve text recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4, dec. 2008.

[10] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR 2010*, pages 2963 –2970, 6 2010.

[11] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, C-22(1):67 – 92, jan. 1973.

[12] E. Kim, S. Lee, and J. Kim. Scene text extraction using focus of mobile camera. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 166 –170, july 2009.

[13] K. Kim, H. Byun, Y. Song, Y. Choi, S. Chi, K. Kim, and Y. Chung. Scene text extraction in natural scene images using hierarchical feature combining and verification. In *Pattern Recognition, 2004. ICPR 2004.*

Proceedings of the 17th International Conference on, volume 2, pages 679 – 682 Vol.2, aug. 2004.

[14] J. Lafferty. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann, 2001.

[15] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch. Adaboost for text detection in natural scene. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 429 –434, sept. 2011.

[16] C. Li, X. Ding, and Y. Wu. Automatic text location in natural scene images. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 1069 –1073, 2001.

[17] L. Li and C. L. Tan. Character recognition under severe perspective distortion. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 –4, dec. 2008.

[18] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *Circuits and Systems for Video Technology*, 12(4):256 –268, 2002.

[19] J. Lim, J. Park, and G. G. Medioni. Text segmentation in color images using tensor voting. *Image and Vision Computing*, 25(5):671 – 685, 2007.

[20] X. Lin. Reliable OCR solution for digital content re-mastering. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Dec. 2001.

[21] S. M. Lucas. Text locating competition results. *Document Analysis and Recognition, International Conference on*, 0:80–85, 2005.

[22] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *ICDAR 2003*, page 682, 2003.

[23] C. Mancas-Thillou and B. Gosselin. Color text extraction with selective metric-based clustering. *Computer Vision and Image Understanding*, 107(12):97 – 107, 2007. ¡ce:title¿Special issue on color image processing¡/ce:title¿.

[24] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22:761–767, 2004.

[25] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *Computer Vision and Pattern*

Recognition (CVPR), 2012 IEEE Conference on, pages 2687 –2694, june 2012.

[26] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *ACCV 2010*, volume IV of *LNCS 6495*, pages 2067–2078, November 2010.

[27] L. Neumann and J. Matas. Estimating hidden parameters for text localization and recognition. In *Proc. of 16th CVWW*, pages 29–26, February 2011.

[28] L. Neumann and J. Matas. Text localization in real-world images using efficiently pruned exhaustive search. In *ICDAR 2011*, pages 687–691, 2011.

[29] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3538 –3545, 6 2012.

[30] A. Newell and L. Griffin. Multiscale histogram of oriented gradient descriptors for robust character recognition. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1085 –1089, sept. 2011.

[31] W. Niblack. *An introduction to digital image processing*. Strandberg Publishing Company, Birkeroed, Denmark, Denmark, 1985.

[32] J. Ohya, A. Shio, and S. Akamatsu. Recognizing characters in scene images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(2):214 –220, feb 1994.

[33] C. Pal, C. Sutton, and A. McCallum. Sparse forward-backward using minimum divergence beams for fast training of conditional random fields. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, page V, may 2006.

[34] Y.-F. Pan, X. Hou, and C.-L. Liu. A robust system to detect and localize texts in natural scene images. In *Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on*, pages 35 –42, sept. 2008.

[35] Y.-F. Pan, X. Hou, and C.-L. Liu. Text localization in natural scene images based on conditional random field. In *ICDAR 2009*, pages 6–10. IEEE Computer Society, 2009.

[36] Y.-F. Pan, X. Hou, and C.-L. Liu. A hybrid approach to detect and localize texts in natural scene images. *Image Processing, IEEE Transactions on*, 20(3):800 –813, march 2011.

[37] M. Sarifuddin. A new perceptually uniform color space with associated color similarity measure for contentbased image and video retrieval. In *in Proc*, pages 3–7, 2005.

[38] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[39] A. Shahab, F. Shafait, and A. Dengel. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *ICDAR 2011*, pages 1491–1496, 2011.

[40] J. Sochman and J. Matas. Waldboost - learning for time constrained sequential detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 150 – 156 vol. 2, june 2005.

[41] H. Takahashi and M. Nakajima. Region graph based text extraction from outdoor images. In *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, volume 1, pages 680 – 685 vol.1, july 2005.

[42] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1457 –1464, nov. 2011.

[43] K. Wang and S. Belongie. Word spotting in the wild. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV 2010*, volume 6311 of *Lecture Notes in Computer Science*, pages 591–604. Springer Berlin / Heidelberg, 2010.

[44] J. Weinman, E. Learned-Miller, and A. Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(10):1733 –1746, oct. 2009.

[45] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1083 –1090, june 2012.

[46] M. Yokobayashi and T. Wakahara. Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 167 – 171 Vol. 1, aug.-1 sept. 2005.

[47] J. Zhang and R. Kasturi. Character energy and link energy-based text extraction in scene images. In *ACCV 2010*, volume II of *LNCS 6495*, pages 832–844, November 2010.