2013 IEEE International Conference on Computer Vision

Scene Text Localization and Recognition with Oriented Stroke Detection

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 unconstrained end-to-end text localization and recognition method is presented. The method introduces a novel approach for character detection and recognition which combines the advantages of sliding-window and connected component methods. Characters are detected and recognized as image regions which contain strokes of specific orientations in a specific relative position, where the strokes are efficiently detected by convolving the image gradient field with a set of oriented bar filters.

Additionally, a novel character representation efficiently calculated from the values obtained in the stroke detection phase is introduced. The representation is robust to shift at the stroke level, which makes it less sensitive to intra-class variations and the noise induced by normalizing character size and positioning. The effectiveness of the representation is demonstrated by the results achieved in the classification of real-world characters using an euclidian nearest-neighbor classifier trained on synthetic data in a plain form. The method was evaluated on a standard dataset, where it achieves state-of-the-art results in both text localization and recognition.

1. Introduction

Detecting and reading text in scene images (also known as *scene text localization and recognition* or photo OCR) is an open problem of computer vision with many interesting applications, ranging from helping visually impaired people, translating language with an automatic input of text written in an unknown script, to indexing large image/video databases by their textual content (e.g. Google Street View, Flickr, etc.). Unlike traditional document OCR, none of the scene text recognition methods has yet achieved sufficient accuracy for practical applications (the winner of the most recent competition achieved a localization recall of only 62% [14]), which is why the problem has been recently receiving significant attention.

Text localization can be computationally very expensive because in an image of N pixels generally any of the 2^N subsets can correspond to text. Text localization methods can be divided into two groups based on how they deal with this issue.

The methods in the first group exploit a sliding-window approach to localize individual characters [17] or whole words [6], drawing inspiration from other object detection methods where this approach has been has been successfully applied [2, 16]. Strengths of such methods include robustness to noise and blur, because they exploit features aggregated over the whole region of interest. The main drawback is that the number of rectangles that needs to be evaluated grows rapidly when text with different scale, aspect, rotation and other distortions has to be found - an effect which does not occur in general object detection tasks where the variance of sliding window parameters is lower.

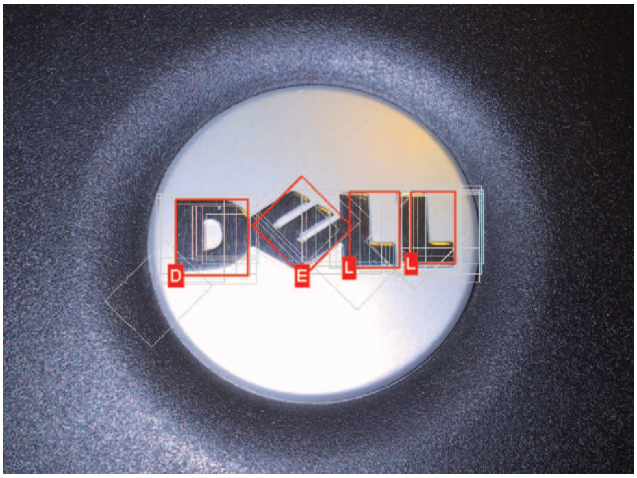

Figure 1. Character detection and recognition of connected and rotated characters. The character representation allows efficient detection of rotated characters, as only a permutation of the feature vector is required. Characters in orientations $0°$, $45°$ and $90°$ were detected and overlapping regions suppressed (marked gray) using classification confidence

The second, recently more popular approach [4, 13, 11, 19, 12, 15] is based on localizing individual characters as connected components using local properties of an image (color, intensity, stroke-width, etc.). The complexity of the methods does not depend on the parameters of the text as characters of all scales and orientations can be detected in one pass and the connected component representation also provides character segmentation which can be exploited in an OCR stage. The biggest disadvantage of such methods is a dependence on the assumption that a character is a connected component, which is very brittle - a change in a sin-

1550-5499/13 $31.00 © 2013 IEEE
DOI 10.1109/ICCV.2013.19

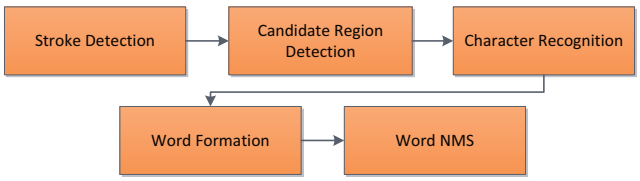

Figure 2. Block structure of the proposed method

gle image pixel introduced by noise can cause an unproportional change in the connected component size, shape or other properties, which subsequently affects its classification. The assumption also prevents the methods from detecting characters which consist of several connected components or where multiple characters are joint into a single connected component.

In this paper, we present an unconstrained end-to-end text localization and recognition method, which detects and recognizes characters as image regions which contain strokes of specific orientations in a specific relative position, where the strokes are efficiently detected by convolving the gradient field with a set of oriented bar filters.

As a first contribution, we introduce a novel approach for character detection which combines the advantages of sliding-window and connected component methods. In the proposed method, the detected strokes induce the set of rectangles to be classified, which reduces the number of rectangles by three orders of magnitude when compared to the standard sliding-window methods. From the complexity perspective this makes the proposed method competitive with the methods based on connected components, but at the same time the robustness against noise and blur is maintained and the assumption that a character is a connected component is dropped, which allows for detection of joint or disconnected characters.

As a second contribution, a novel character representation efficiently calculated from the values obtained in the stroke detection phase is introduced. The representation is robust to shift at the stroke level, which makes it less sensitive to intra-class variations and the noise induced by normalizing character's size and positioning. The effectiveness of the representation is demonstrated by the results achieved in classification of real-world characters using a linear (approximative) nearest-neighbor classifier trained on synthetic data in a plain form (i.e. without any blurring or distortions). Additionally, the representation allows for efficient detection of rotated characters, as only a permutation of the feature vector is required (see Figure 1).

The method was evaluated on the most cited dataset [14], where it achieves state-of-the-art results in both text localization and recognition.

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

2. Previous Work

Several methods which focus only on a particular subproblem (text localization [6, 13, 19, 4] or character respectively word cut-out recognition [3, 9]) have been published. Most relevantly, the method of Epstein et al. [4] converts an input image to a greyscale and uses the Canny detector [1] to find edges. Pairs of parallel edges are then used to calculate stroke width for each pixel and pixels with a similar stroke width are grouped together into characters. Alongside the aforementioned limitation of the connected components methods, the method also relies on a successful edge detection which might be problematic in noisy images and moreover it cannot handle ambiguities because each image pixel can belong to only one stroke. The proposed method differs radically from [4] in that it does not rely on hard decisions made by an edge detector, and in that it does not aim to estimate the stroke width (it actually assumes a unit stroke width - see Section 3.1), but rather it estimates the possible positions of strokes and detects characters based on known patterns of stroke orientations and their relative positions. The method of Epstein et al. provides character segmentation but does not perform text recognition.

The method of Wang and Belongie [17] 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 and the sliding-window is limited to horizontal rectangles with a limited number of scales and aspects.

An end-to-end text localization and recognition method [12] introduced by Neumann and Matas detects characters as a subset of Extremal Regions and then recognizes candidate regions in a separate OCR stage. The method performs well even on low contrast images and its computational complexity is close to real time. The text recognition however performs poorly on noisy images, because of the sensitivity induced by the connected component assumption. Moreover, the character representation exploited in the method [11] is based on a direction of the boundary pixels chain-code, whose robustness is limited. For an exhaustive survey of text localization and recognition methods refer to the ICDAR Robust Reading competition results [8, 7, 14].

3. The Proposed Method

We assume that each character is defined by a set of its strokes and their relative position. For instance, the letter "F" consists of two strokes in the 0° direction and one stroke in the 90° direction, where the 90° stroke is to the left from the two 0° strokes and on the contrary the 0° strokes are lo-

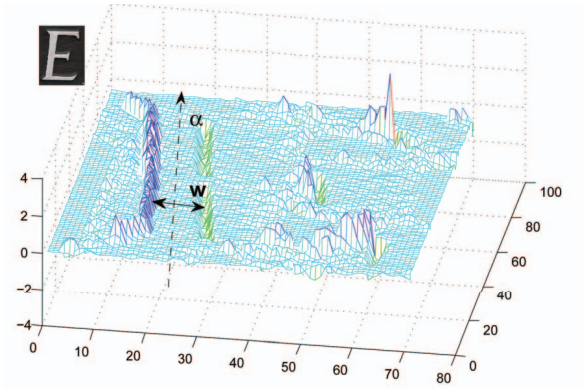

Figure 3. A stroke of direction α is detected as two opposing ridges in the gradient (approximately) perpendicular to the stroke direction. Note that the distance w between the ridges is the *stroke width*

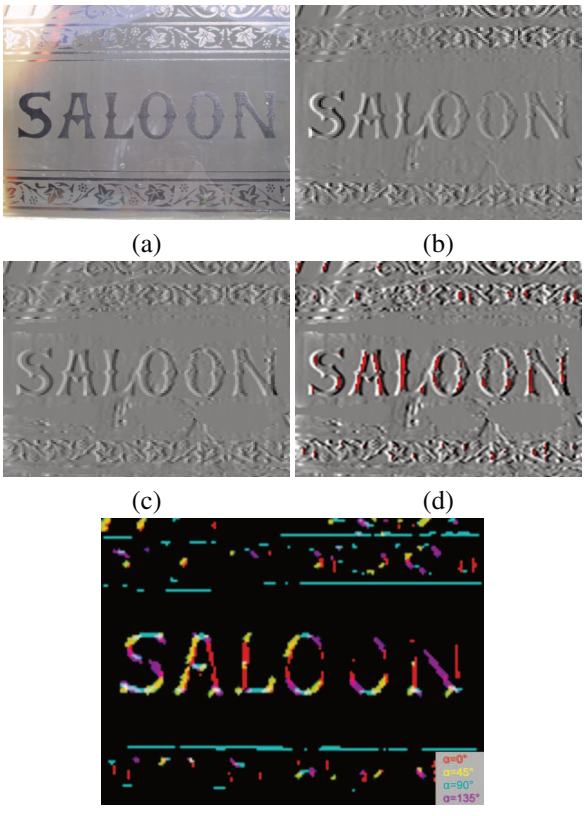

Figure 4. Oriented stroke detection. Source image (a). Image gradient projection in the direction $\alpha = 0°$ and scale $s = 0.25$ (b). Normalized gradient projection ($G_{0,0.25}$) (c). Response of the convolution filter ($R_{0,0.25}$), normalized gradient projection as a background (d). Color-coded responses of all filters at the scale $s = 0.25$ (e)

cated near the middle respectively the end of the 90° stroke.

In the proposed method, the strokes are modelled as responses to oriented filters in the gradient projection scale space (see Section 3.1) and the relative stroke position is modelled by subsampling the responses into a fixed-sized matrix. Characters are detected by recognizing a known stroke pattern with a classifier trained with synthetic data (see Section 3.3).

3.1. Stroke Detection

Let us consider an image $\mathbf{I}$ as a mapping $\mathbf{I} : \mathcal{D} \subset \mathbb{N}^2 \rightarrow [0, 1]$. A gradient projection $G_{\alpha,s}$ in the direction α and scale s is the change of intensity in the image $\mathbf{I}$ rotated by angle α and resized to the scale s, taken in the horizontal direction, i.e.

$$G_{\alpha,s} = \frac{\partial R_\alpha S_s \mathbf{I}}{\partial x} \tag{1}$$

where R_α denotes a rotation matrix of an angle α and S_s denotes a scaling matrix of a scale s.

A stroke of direction α can be detected as two opposing ridges in the gradient perpendicular to the stroke direction (see Figure 3), where the distance w between the two ridges corresponds to *stroke width*. In the proposed method, we assume the stroke width value is one ($w = 1$) and we search for all strokes of unit width in a scale space by convolving the gradient projection with a 5×5 filter that responds to such strokes.

The response of the convolution filter $R_{\alpha,s}$ in the direction α and scale s is defined as

$$\begin{aligned} R_{\alpha,s} = \big\{ & G_{\alpha,s} * F_{-1,1} - \max(G_{\alpha,s} * F_{3,1}, 0) \\ & - \max\left(G_{\alpha,s} * F_{-1,-3}, 0\right) \big\}_\Theta \end{aligned} \tag{2}$$

$$F_{a,b} = \begin{pmatrix} a & a & 0 & b & b \\ a & a & 0 & b & b \\ a & a & 0 & b & b \\ a & a & 0 & b & b \\ a & a & 0 & b & b \end{pmatrix} \tag{3}$$

$$\{x\}_\Theta = \begin{cases} x & x \geq \Theta \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where the first term in the Eq. 2 responds to a negative gradient ridge in the distance of one pixel from a positive gradient ridge and the second and third term suppress the response where there is only positive respectively negative gradient ridge.

The thresholding parameter Θ represents a trade-off between an ability to detect strokes of low-contrast and the number of candidate regions to classify (see Section 3.2). In our implementation, we set $\Theta = 8$ for all directions and scales; lowering the threshold did not further improve method's recall, but it increased the number of candidate regions (see Figure 6).

In our experiments, we first normalized the contrast of the gradient projection with a low-pass filter and then convolved it over a range of 10 exponentially decreasing scales (in the interval of 0.05 and 1.0) and 4 orientations (0°, 45°, 90° and 135°) - see Figure 4. The convolution is preformed

Figure 5. Candidate regions are induced through bounding-boxes of strokes, which reduces the number of target rectangles by three orders of magnitude when compared to the sliding-window methods. Candidate regions T (top). Sample image regions extracted for better clarity (bottom). *Note that the proposed method does not extract any image patches as part of the process*

twice, once in the original image and once in an inverted image to detect strokes with an opposite ridge orientation.

3.2. Candidate Region Detection

In the next step, we generate candidate image regions (in the form of bounding-boxes) for classification. Unlike sliding-window methods which exhaustively evaluate all image regions, we exploit the fact that we are only interested in image regions which contain at least one stroke (in our character representation, regions without any stroke would be rejected as non-characters anyways). Moreover, if we assume that for each character there exists a subset of its strokes that induces its bounding-box, we can efficiently generate candidate regions by taking unions of stroke bounding-boxes.

Let us denote $\mathcal{B}_{\alpha,s}$ as the set of bounding-boxes of connected components (strokes) in $R_{\alpha,s}$ (where the binarization was obtained by simply taking non-zero values). The set of candidate regions T is then defined as

$$T = \bigcup_{b \in \mathcal{B}} \bigcup_{k=0}^{K} \upsilon(c,k) \tag{5}$$

$$\mathcal{B} = \bigcup_{\forall \alpha,s} \mathcal{B}_{\alpha,s} \tag{6}$$

$$\upsilon(c,k) = \cup \{b, N_1(b), \ldots, N_k(b)\}\,,\ N_k(b) \in \mathcal{B} \tag{7}$$

where $\cup$ denotes a union of bounding-boxes (the smallest rectangle that contains all rectangles in the set), $\bigcup$ is a standard union of sets and $N_k(b)$ is the k-th nearest component from the bounding-box b (measured by distance of the bounding-boxes' centers). In other words, for each connected component (stroke) we consider its bounding-box and then K bounding-boxes created as a union of bounding-boxes of 1 to K nearest connected components (see Figure 5).

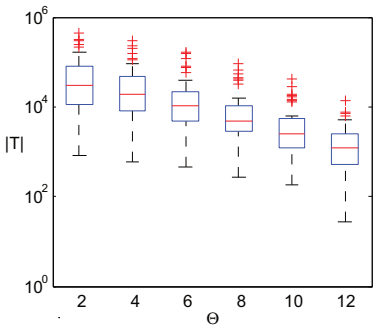

Figure 6. The number of candidate regions $|T|$ as a function of the thresholding parameter Θ

The problem is overcomplete as typically there are many different combinations of strokes which induce an identical or nearly identical character bounding-box, which reduces the probability of missing a true bounding-box in such a greedy approach. For example, consider the letter "E" - it consists of 4 strokes (3 horizontal and 1 vertical) and 4 out of 6 possible stroke pairs (and all 4 possible stroke triplets) induce identical character bounding-box. Moreover, the exact position of the character bounding-box is not crucial because the character representation is robust to shift.

This property also allows to further improve the method's performance by eliminating similar rectangles from the T set by keeping only the largest rectangle of the similar rectangles (two rectangles are considered similar if their intersection is more than 95% of their union). Rectangles smaller or larger than a size of the smallest resp. largest character to be detected are also removed.

On average the number of candidate regions in an image is of the order of magnitude between 10^3 and 10^4 (depending on the amount of text and other stroke-like textures), which is orders of magnitude less than the number of candidate regions in a standard exhaustive sliding-window approach (in an image of 2×10^6 pixels, there are $\approx 2 \times 10^6$ different window locations times number of scales and aspects, i.e. magnitude between 10^6 and 10^8).

The number of neighboring bounding-boxes K was set to 5; increasing the value further had very little impact on the overall results because of the overcompletness of the task and also because characters in our datasets (coming from Latin alphabet) consist of a relatively low number of strokes. Applying this method to a different script (e.g. Chinese script) might however require an increase of the parameter value, but this still would be computationally feasible as the number of candidate regions is linear in the number of strokes.

3.3. Character Recognition

Each candidate region $b \in T$ is labelled with a Unicode code(s) or rejected as "unknown" in the following process. At first, a response of the candidate region $R_\alpha(b)$ is calculated as a maximum pooled over an interval of scales

$$R_\alpha(b) = \overline{\max}_{s \in \rho(b)} M_{20}(R^b_{\alpha,s}) \quad (8)$$

$$\overline{\max}(X^1, X^2, \ldots, X^L) = \left(m_{i,j}\right) : m_{i,j} = \max_{l=1}^{L} x^l_{i,j} \quad (9)$$

$$M_n(X_{d\times d}) = \left(m_{i,j}\right)_{n\times n} :$$
$$m_{i,j} = \max_{k=1}^{S} \max_{l=1}^{S} x_{Si+k,Sj+l}$$
$$S = \frac{d}{n} \quad (10)$$

where $\overline{\max}$ is an element-wise maximum, $\rho(b)$ is the scale interval, $M_n(X)$ is a pooling operation which extracts maximums of X into a fixed-size matrix of size n and R^b is the subregion of R corresponding to the bounding-box b. If the region is not a square, it is padded with zeros to form a squared matrix.

Only a subset of scaled responses which were precomputed in the first stage (see Section 3.1) is used for each region, depending on its size and aspect, so that strokes from lower scales do not suppress the ones from a higher scale. The subset is determined by the trained function $\rho(b)$, which maps region's height and width to an interval of admissible scales. For example, a region which is two times wider than higher can only be occupied by characters with a similar aspect (i.e. "m", "w", ...) and this limits the interval of possible stroke widths. Because of the assumption of the unit stroke width, the interval of possible stroke widths unambiguously determines the interval of admissible scales.

The response in each orientation $R_\alpha(b)$ is then blurred with a 7×7 Gaussian filter ($\sigma = 2$) for partial shift-invariance and subsampled again with a M_5 operation to form a 5×5 matrix $\hat{R}_\alpha$.

$$\hat{R}_\alpha = M_5\left(R_\alpha(b) * G^2_{7\times 7}\right) \quad (11)$$

The final feature vector $f(b)$ is created by concatenating responses of all 4 orientations $\hat{R}_\alpha(b)$, thus forming a 100-dimensional feature vector (see Figure 7).

Given a training set $\mathcal{T}$, a set of K nearest neighbors $N(b) = \mathcal{N}_K(f(b)), N(b) \in \mathcal{T}$ is assigned to each candidate region (see Figure 9). A set of admissible labels $\hat{L}(b)$ of a region b is then defined as

$$\hat{L}(b) = \left\{l(n) : n \in N(b) \mid \|f(n) - f(b)\| \leq \bar{d}(l(n))\right\} \quad (12)$$

where $l(n)$ denotes label of the training sample n and $\bar{d}(l)$ is a maximal distance for the label (class) l. A set of character regions $\mathcal{R}$ is then defined as $\left\{b \in T \mid \hat{L}(b) \neq \emptyset\right\}$.

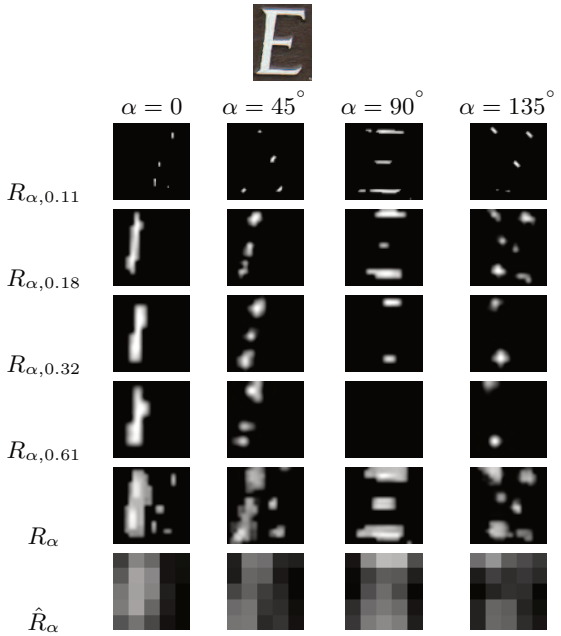

Figure 7. The character representation is based on positions of oriented strokes, which are pooled over multiple scales

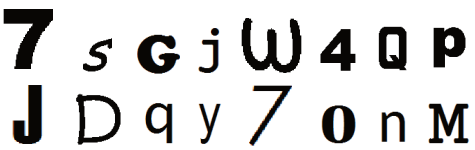

Figure 8. Random samples from the training set. The set contains 5580 characters from 90 fonts with no distortions, blurring or rotations

In our experiments, the training set consists of images with a single black letter on a white background (see Figure 8). In total there were 5580 training samples (62 character classes in 90 different fonts). Let us note that no further distortions, blurring, scaling or rotations were artificially introduced to the training set, in order to demonstrate the power of the feature representation, and that the method can be easily extended to incorporate additional characters or even scripts (without any limitation on the number of classes) and to detect fonts with almost no overhead.

The nearest-neighbor classifier $\mathcal{N}_K$ was implemented by an approximative nearest-neighbor classifier [10] for performance reasons and K was set to 11. The values $\bar{d}(l)$ were estimated for each class by a cross-validation on the training set as an average maximal distance over all folds, multiplied by a tolerance factor of β. The value of β represents a trade-off between detecting more characters from fonts not in the training set and more false positives. In our experiments, we used the value $\beta = 2.5$, which yields the best performance on the training subset of the ICDAR dataset (see Section 4).

3.4. Word Formation

Given a set of character regions $\mathcal{R}$, the regions are agglomerated into a set of text lines $\mathbf{T}$ (see Algorithm 1). A

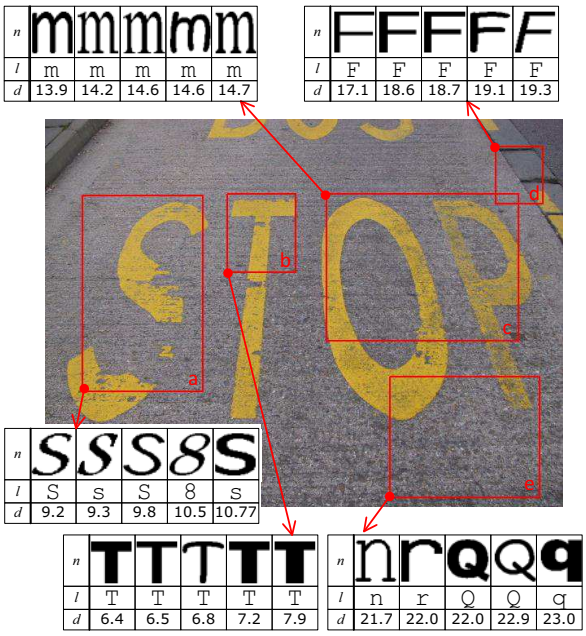

Figure 9. Candidate regions are classified with a nearest-neighbor classifier trained on synthetic data. Note that the distance from the templates is not dramatically affected when the letter is disconnected (a) and that letter-like regions (c, d) have lower distance from templates and lower entropy of labels than regions without characters (e)

text line $T \in \mathbf{T}$ is a partially ordered set of regions whose bottom points are arranged in a single direction. The partial ordering is induced by relative position of the regions in the direction of the text line and represents a left-to-right ordering of characters in a word. In other words, the partial ordering is induced by the restriction that a region can only be preceded by regions to the left and succeeded by regions to the right from the particular region, allowing for a small overlap.

To detect words in the image and recognize their content, an optimal sequence is found in each text line (where the order in the sequence is induced by the partial ordering of the text line) by maximizing the objective function

$$\mathcal{L}^*(T) = \text{argmax}_{\forall i, l \in \hat{L}(r_i)} \Big(\sum_{i=1}^{|T|} \phi(r_i, \hat{L}(r_i)) + \\ + \sum_{i=1}^{|T|-2} \psi\left(r_i, r_{i+1}, r_{i+2}\right) \Big), r_i \in T \quad (13)$$

$$\phi(r, l) = \log p_L(l|r) \quad (14)$$

$$\psi(r_i, r_{i+1}, r_{i+2}) = \log p_S(r_{i+2}|r_i, r_{i+1}) p_P(r_{i+2}|r_i, r_{i+1}) \\ p_A(r_{i+2}|r_i, r_{i+1}) \quad (15)$$

where p_L is the probability of the region r having the label l, p_S is the probability of the inter-character spacing difference, p_P is the probability of regions' relative positioning and p_A is the adjacency probability given by the language model.

The probability p_L is approximated as a weighted sum of the relative distances of the K nearest neighbors (see Section 3.3)

$$p_L(l|r) \approx \frac{1}{K} \sum_{n \in N(r): l(n)=l} \frac{\min_{n' \in N(r)} d(n', r)}{d(n, r)}$$

$$d(n, r) = \|f(n) - f(r)\| \quad (16)$$

The probability p_S models the observation that the spacing between characters does not vary a lot in a single word. We define the difference of spacing Δs of three regions as

$$\Delta s(r_1, r_2, r_3) = \frac{|s_{12} - s_{23}|}{\max(s_{12}, s_{23})}$$

$$s_{ij} = r_j^L - r_i^R \quad (17)$$

where r^L and r^R denote left respectively right boundary of the region in the orientation of text. The probability p_S is then estimated from Δs, assuming that Δs is a random variable with a Normal distribution (whose parameters were obtained on synthetically generated words of different fonts).

Similarly, the probability p_P models the observation that positioning of a character triplet is not arbitrary. We define the difference of top Δt respectively bottom Δb positions as

$$\Delta t(r_1, r_2, r_3) = \min\left(|r_1^T - r_2^T|, |r_2^T - r_3^T|, |r_1^T - r_3^T|\right) \quad (18)$$

$$\Delta b(r_1, r_2, r_3) = \min\left(|r_1^B - r_2^B|, |r_2^B - r_3^B|, |r_1^B - r_3^B|\right) \quad (19)$$

where again r^T and r^B denote top respectively bottom boundary of the region. The probability p_S is then estimated from Δt, whose distribution is modelled by a two-dimensional Gaussian Mixture Model.

The probability p_A is approximated by relative frequencies of character triplets, which are calculated in the training stage (a list of approx. 30000 English words was used to generate the relative frequencies).

A standard Dynamic Programming approach is used to find the optimal value of Equation 13 (because the function contains 2nd order terms, it is first necessary to transform the task into a 1st order problem by expanding the state space) and the optimal sequence is selected as content of the text line. As a final step, spaces are detected as peaks in the histogram of inter-character spacings to break down text lines into words and overlapping words are eliminated through a non-maximum suppression.

4. Experiments

The proposed method was evaluated on the ICDAR 2011 Robust Reading competition dataset [14], which contains 1189 words and 6393 letters in 255 images. Using

Data: a set of regions $\mathcal{R}$
Result: a set of text lines $\mathbf{T}$
$\mathbf{T} \longleftarrow \emptyset$;
$R \longleftarrow \mathcal{R}$;
$D \longleftarrow [-45, -35, \ldots, 0, \ldots, 45]$;
while $|R| > 0$ **do**
  for $d \in D$ **do**
    $B \longleftarrow$ bottom points of $r \in R$;
    rotate $b \in B$ by d;
    $H_d \longleftarrow$ histogram of y-coordinates of $b \in B$;
  end
  $\bar{d} \longleftarrow$ index of the histogram H_d with the greatest peak;
  $y \longleftarrow$ the value corresponding to the greatest peak in $H_{\bar{d}}$;
  $l \longleftarrow$ line in the direction $\bar{d}$ and position y;
  $T \longleftarrow \{r \in R \ : \$ the bottom point of r is close to the line $l\ \}$;
  $R \longleftarrow R \setminus T$;
  $\mathbf{T} \longleftarrow \mathbf{T} \cup T$;
end

Algorithm 1: Text line formation

the ICDAR 2011 competition evaluation protocol [18], the method reaches the recall of 66.4%, precision of 79.3% and the f-measure of 72.3% in text localization (see Figure 10 for sample outputs).

The method achieves significantly better recall (66%) than the winner of ICDAR 2011 Robust Reading competition (62%) and the recently published Shi's method [15] (63%). The overall f-measure (72%) outperforms all published methods (see Table 1), but the precision is worse than the winner of the ICDAR competition. Let us 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.

In the end-to-end text recognition, the method achieves the recall of 45.4%, the precision of 44.8% and the f-measure of 45.2%, which significantly outperforms the results achieved by the recently published method of Neumann and Matas [12] (see Table 2).

The average processing time of the code implemented in Matlab is approx. 35s per image on a standard PC. The processing time can be further improved by parallelization and a more efficient implementation of the convolution and dynamic programming stages.

The problems of the method include ambiguities introduced by the fact that a subregion of a character might be another character, failures to detect letters on word boundaries which consist of just one stroke (e.g. "I", "l") and false positives caused by strokes around areas with text (see Figure 11).

method	recall	precision	f
proposed method	**66.4**	**79.3**	**72.3**
Shi et al. [15]	63.1	83.3	71.8
Kim (ICDAR'11 winner) [14]	62.5	83.0	71.3
Neumann and Matas [12]	64.7	73.1	68.7
Epshtein et al. [4]	60.0	73.0	66.0
Yi's Method [20]	58.1	67.2	62.3
TH-TextLoc System [5]	57.7	67.0	62.0

Table 1. Comparison with most recent text localization results on the ICDAR 2011 dataset.

method	recall	precision	f
proposed method	**45.4**	**44.8**	**45.2**
Neumann and Matas [12]	37.2	37.1	36.5

Table 2. Comparison with most recent end-to-end text recognition results on the ICDAR 2011 dataset.

5. Conclusions

An end-to-end real-time text localization and recognition method was presented in the paper. The method introduces a novel approach for character detection and recognition which combines the advantages of sliding-window and connected component methods. Characters are detected and recognized as image regions which contain strokes of specific orientations in a specific relative position, where the strokes are efficiently detected by convolving the image gradient field with a set of oriented bar filters. The characters are selected from an efficiently obtained set of target regions by a nearest-neighbor classifier, which exploits novel character representations based on strokes.

On the standard ICDAR 2011 dataset [14], the method achieves state-of-the-art results in both text localization and end-to-end text recognition.

Acknowledgment

The authors were supported by the EU project MASELTOV (FP7-ICT-288587), CTU Grant Agency project SGS13/142/OHK3/2T/13 and Technology Agency of the Czech Republic program TE01020415 (V3C - Visual Computing Competence Center). Lukas would also like to acknowledge the Google PhD Fellowship and the Google Research Award.

References

[1] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986. 2

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR 2005*, volume 1, pages 886–893. IEEE, 2005. 1

[3] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. *VISAPP*, 05-08 February 2009, 2009. 2

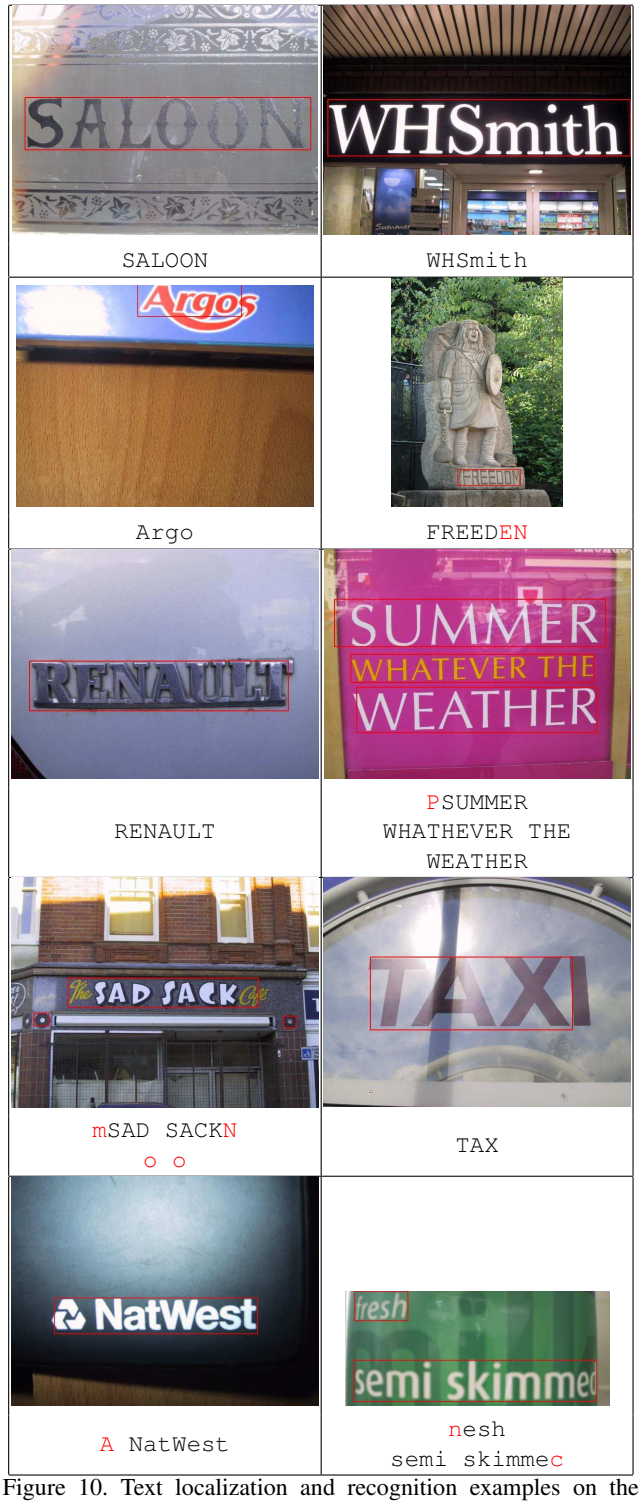

Figure 10. Text localization and recognition examples on the ICDAR 2011 dataset

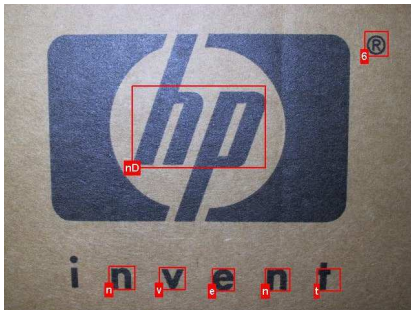

Figure 11. Problems of the proposed method. Ambiguities introduced by the fact that a subregion of a character might be another character ("nD"). A failed detection of a letter on word boundary which consists of just one stroke ("i")

[4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR 2010*, pages 2963 –2970. 1, 2, 7

[5] S. M. Hanif and L. Prevost. Text detection and localization in complex scene images using constrained adaboost algorithm. In *ICDAR 2009*, pages 1–5. IEEE, 2009. 7

[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. 1, 2

[7] S. M. Lucas. Text locating competition results. *ICDAR 2005*, 0:80–85, 2005. 2

[8] 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. 2

[9] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *CVPR 2012*, pages 2687 –2694, june 2012. 2

[10] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISSAPP09*, pages 331–340, 2009. 5

[11] 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. 1, 2

[12] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *CVPR 2012*, pages 3538 –3545, 6 2012. 1, 2, 7

[13] 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. 1, 2

[14] 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. 1, 2, 6, 7

[15] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao. Scene text detection using graph model built upon maximally stable extremal regions. *PR*, 34(2):107 – 116, 2013. 1, 7

[16] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 1

[17] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV 2011*, 2011. 1, 2

[18] 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. 7

[19] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR 2012*, pages 1083 –1090, june 2012. 1, 2

[20] 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. 7