Efficient character skew rectification in scene text images

Michal Bušta, Tomáš Drtina, David Helekal, Lukáš Neumann, Jiří Matas

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

Abstract. We present an efficient method for character skew rectification in scene text images. The method is based on a novel skew estimators, which exploit intuitive glyph properties and which can be efficiently computed in a linear time. The estimators are evaluated on a synthetically generated data (including Latin, Cyrillic, Greek, Runic scripts) and real scene text images, where the skew rectification by the proposed method improves the accuracy of a state-of-the-art scene text recognition pipeline.

1 Introduction

Scene text detection and recognition is an open problem with many interesting applications. When compared to traditional printed document OCR (which is considered to be a solved problem), scene text recognition faces additional challenges such as complex backgrounds, variations of font face, size, texture and image distortions. The distortions may include perspective distortion, blur distortion, motion blur or a combination of any of the above.

In this paper, we focus on the problem of text skew estimation and subsequential rectification, in order to improve text recognition accuracy. We refer to *skew* as the angle of individual letters to the text direction, whereas we refer to *rotation* as orientation of a word baseline relative to the horizontal direction (see Figure 1).

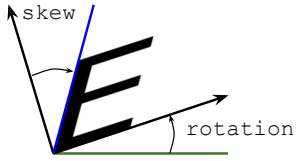

Fig. 1. Character skew and rotation.

We propose five novel character skew estimators, which exploit intuitive glyph properties and which can be efficiently computed in a linear time. The estimators are first evaluated on a synthetically generated data and then an existing

TextSpotter pipeline ([1,2]) is adapted to evaluate the impact of character skew rectification to the scene text recognition accuracy.

The rest of the paper is structured as follows. In Section 2 previous work is presented, in Section 3 the problem of scene text geometry is introduced, in Section 4 the proposed method is described and in Section 5 the experimental validation is given. The paper is concluded in Section 6.

2 Related Work

The most common way to estimate text skew is using the projection profiles [3,4]. In these methods, text skew is iterated in small increments and for each value a vertical projection profile is calculated (see Figure 2). The projection profile looks like a series of peaks and valleys, which correspond to spaces between characters; for the correct skew the peaks of the vertical projection profile are going to be the highest. This can be computationally expressed by a number of measures, such the entropy of the histogram, standard deviation variation or sum of squares. The major drawback of these methods is their computational complexity, since the interval of possible skews has to be iterated in small increments (this may even prohibit the use of such methods on devices with lower performance, such as mobile phones).

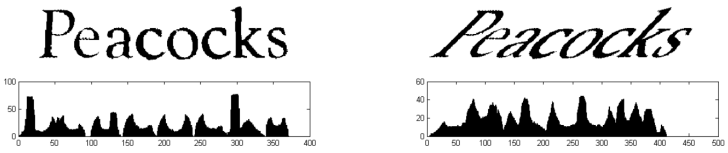

Fig. 2. In previous work, text skew is estimated by iterating all skew values and selecting an optimal value using vertical projection profiles.

Another group of methods focuses on skew estimation for whole documents. The methods are based on the Hough transform [5,6], distance transform [7], gradient Directions [8] and Focused Nearest Neighbor Clustering (FNNC) of interest points [8]. The main limitation of these methods is the fact that they require a whole document with a lot of text in a binarized form to estimate the skew, which is often unusable for scene text where binarization (segmentation) may not be as accurate and which usually consists of short text snippets, sometimes even with different skew or orientation.

3 Scene Text Geometry

Assuming that scene text typically appears on planar surfaces (see Figure 3), the text transformation in an image can be modelled in the standard pinhole camera model as a homography.

Fig. 3. Scene text typically appears on planar surfaces.

Let us recall that a homography $\mathbf{H}$ can be decomposed as

$$\mathbf{H} = \mathbf{EASP} = \begin{pmatrix} s\cos(\theta) & s\sin(\theta) & t_x \\ -s\sin(\theta) & s\cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/b & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\tan(\alpha) & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_x & l_y & 1 \end{pmatrix} \tag{1}$$

where $\mathbf{E}$ is a 2D rigid transformation, $\mathbf{A}$ is an anisotropic scaling transformation, $\mathbf{S}$ is a skew transformation and $\mathbf{P}$ is a projective transformation.

A homography has 8 degrees of freedom: rotation θ, isotropic scaling factor s, skew α, anisotropic scaling factor b, translation factors t_x, t_y, and perspective shortening l_x and l_y.

For character recognition, the scaling and translation factors s, b, t_x and t_y are not important since in the OCR stage character regions are typically normalized to a fixed size (this applies to both training and recognition time).

Exploiting the fact that in a typical character detection pipeline text lines are formed before the character recognition stage (see Figure 4), the direction of the text (i.e. the text bottom line direction) can be used as an estimate for the rotation θ.

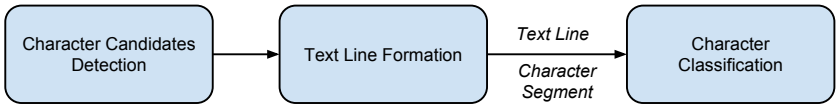

Fig. 4. In typical scene text recognition pipelines the text direction is known before the character recognition stage.

If we assume that the effect of perspective shortening is not significant, the only remaining parameter required for text rectification is the skew α. The skew α of a text in an image depends on the position of the plane in the scene, but also on the topographical parameters of the font (e.g. a text can be written in

italics). The skew α may therefore be different for different words on the same plane and it is beneficial to estimate skew for each word individually.

In this paper, we focus on estimating the skew α from word segmentations (which may contain only few characters). Applying a reverse transformation to rectify the text skew (and rotation) improves OCR accuracy and helps to significantly reduce the size of the training set as fonts with different slanting (e.g. italics) don't have to be included.

Fig. 5. Rotation rectification is not sufficient for scene text images as text is typically perspectively distorted. Scene text image (left). Rotation θ rectified (top right). Rotation θ and skew α rectified (bottom right).

Let us also note that omitting skew estimation can lead to problems, since text in scene images is often perspectively distorted and rectifying only rotation leaves the character segmentations slanted (see Figure 5).

4 Skew Estimation

In this paper, we propose five novel skew estimators for character segmentation or its polygonal approximation (the Ramer-Douglas-Peucker algorithm [9] was used to calculate the boundary approximation).

The estimators are based on intuitive character glyphs features (e.g. character symmetricity, dominant stroke, etc.), are easy to implement and have a low computational cost (most of them are linear in number of character glyph points).

4.1 Vertical Dominant (VD)

Assumption 1 *The dominant tangent direction of an undistorted character is vertical.*

In other words, the dominant tangent direction corresponds to the skew α (see Figure 6). To find the dominant tangent direction, each edge of the polygonal approximation contributes in its direction with a weight proportional to its length to a Parzen-window density estimate

$$P(\alpha) = \frac{1}{N} \sum_{i=0}^{N} \frac{l_i}{\delta\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\alpha - \alpha_i}{\delta}\right)^2\right) \tag{2}$$

where N is the number of edges in approximated contour, l_i is the edge length, α_i is the edge angle and δ ($\delta = 3°$) is the standard deviation of the Gaussian PDF (smoothing parameter).

The skew estimate $\hat{\alpha}$ then corresponds to the maximum of the density estimate

$$\hat{\alpha} = \arg\max(P(\alpha)) \tag{3}$$

where the maximal value of $P(\alpha)$ gives a confidence estimate.

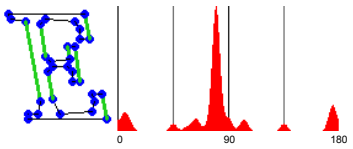

Fig. 6. Vertical Dominant (VD). Character polygonal approximation, edges with the dominant tangent direction highlighted (left). Weighted Parzen-window density estimate $P(\alpha)$ (right).

4.2 Vertical Dominant on Convex Hull (VC)

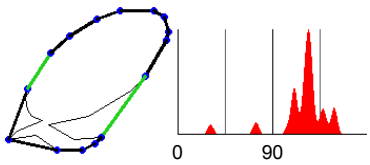

Fig. 7. Vertical Dominant on Convex Hull (VC). Character convex hull, edges with the dominant tangent direction highlighted (left). Weighted Parzen-window density estimate $P(\alpha)$ (right).

It is expected (and verified on the training dataset), that the Vertical Dominant (VD) estimator will perform poorly on characters such as `S`, `7`, where the

dominant tangent angle direction is not vertical. However, if the dominant tangent direction is found on a convex hull instead of the segmentation polygonal approximation, a more accurate estimate can be obtained for some characters (see Figure 7).

4.3 Longest Edge (LE)

Assumption 2 *The longest edge of a character is usually vertical.*

The skew estimate $\hat{\alpha}$ is the direction of the longest edge in the polygonal approximation (see Figure 8).

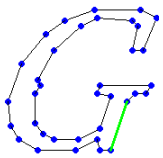

Fig. 8. Longest Edge (LE). Character polygonal approximation and the longest edge highlighted.

4.4 Thinnest Profile (TP)

Assumption 3 *The angle of the thinnest profile of a character corresponds to the skew angle.*

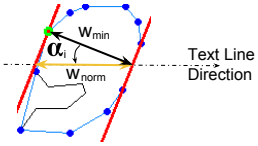

Fig. 9. Thinnest Profile (TP). Character polygonal approximation, its convex hull and the pair of parallel lines with the shortest distance. The thinnest profile w_{min} is normalized using the angle α_i between the thinnest profile and the text line direction

The thinnest profile is found by the Rotating Calipers algorithm [10] on the convex hull of the character polygonal approximation. The width of the profile is normalized using the angle of the measured profile $w_{norm} = w_{min}/\cos\alpha_i$ where α_i is the angle of current profile (see Figure 9). The Rotating Calipers algorithm exploits an analogy of a rotating spring-loaded vernier caliper - every time one blade of the caliper lies flat against an edge of the polygon, the algorithm forms an antipodal pair with the point or edge touching the opposite blade. The

complete "rotation" of the caliper around a convex polygon finds all antipodal pairs and is carried out in $O(N)$ time (where N is the number of convex hull contour points). In our implementation, Sklansky's algorithm [11] is used to form the convex hull from the polygonal approximation, whose complexity is linear in the number of contour points.

4.5 Symmetric Glyph (SG)

Assumption 4 *The top and the bottom parts of a character are vertically symmetric.*

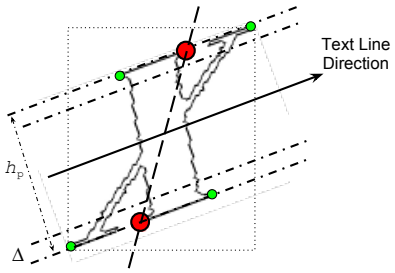

Fig. 10. Symmetric Glyph (SG). The most left and the most right points (green), the top and bottom centers (red).

First, the topmost and the bottom-most points are found as the pixels with longest distance from the center of bounding box of letter glyph in the direction perpendicular to the text line direction. Next, the leftmost and the rightmost pixels are found in the band of the height Δ, which is placed around the topmost (respectively the bottom-most point) and which is parallel to the text line direction. Finally, the top (respectively the bottom) center is found as the center of the line passing trough the most left and the most right points on the top (the bottom) of the character - see Figure 10. Estimated skew $\hat{\alpha}$ is then angle of the line connecting the top and bottom center points.

In our experiments, we set $\Delta = 0.08h$, where h is the height of the character.

We have observed that if the process of skew estimation using the Symmetric Glyph property is repeated (i.e. the region is rectified using the estimated skew $\hat{\alpha}$ and a new estimate is created for the rectified region), the estimation accuracy is improved. Therefore, in our experiments, the process of estimating and rectification is repeated until convergence.

5 Experiments

A testing set of 10 randomly selected fonts was first created to tune the parameters of the skew estimators. The optimal values of each paramater have been selected by a simple grid-search.

5.1 Synthetic Characters

In the first experiment, characters from 5 different scripts (Latin, Cyrillic, Georgian, Greek and Runic) and 50 random fonts for each script (the selected fonts were not present in the training set) were distorted with a random skew α_{gt}. We then compared the difference of the skew estimate $\hat{\alpha}$ of each skew estimator with the ground truth skew α_{gt}. If the estimated skew differs less than 3° we consider the estimate as correct (see Table 1). The standard deviation of the angle difference is measured as well. At last but not least, skew estimation error was broken down by ground truth skew value to see how well small or large distortions are estimated by each estimator (see Figure 11).

Table 1. Skew estimation accuracy a and standard deviation σ on synthetic characters.

	Cyrillic		Georgian		Greek		Latin		Runic		All	
	a[%]	σ[°]	a[%]	σ[°]	a[%]	σ[°]	a[%]	σ[°]	a[%]	σ[°]	a[%]	σ[°]
VD	73.5	10.8	69.1	12.3	67.8	12.7	64.6	14.0	79.7	14.3	68.7	12.7
VC	70.2	9.5	54.5	12.8	61.7	10.8	64.9	8.6	59.2	13.4	66.1	9.5
LE	67.2	12.9	62.0	16.6	58.5	14.9	57.8	15.6	78.7	14.5	61.7	14.5
TP	74.1	5.9	67.2	6.3	68.3	6.5	68.8	6.0	68.3	8.5	70.7	6.1
SG	51.0	11.1	40.1	10.4	61.2	8.0	58.6	8.1	62.7	8.5	56.1	9.4

It can be observed that the most informative is the Thinnest Profile (TP) estimate, which has the highest accuracy and amongst all scripts. On contrary, the Symmetric Glyph (SG) estimate performs the worst on the Cyrillic and Georgian scripts, but is the third best on the Greek script (which suggests that Greek glyphs are more symmetric) and it has the second most consistent estimates for both small and large skews.

In the second experiment, we evaluate how much the estimators are complementary to each other. In Figure 12 we show the density plot of skew estimation error for each pair. From the plot it can be observed that the Vertical Dominant on Convex Hull (VC), the Symmetric Glyph (SG) and the Thinnest Profile (TP) estimators complement each other.

5.2 Synthetic Text

In this experiment, synthetic images where generated for a random subset of dictionary words using a font randomly selected from the Google Fonts directory (which contains approximately 1200 fonts). For each randomly generated word,

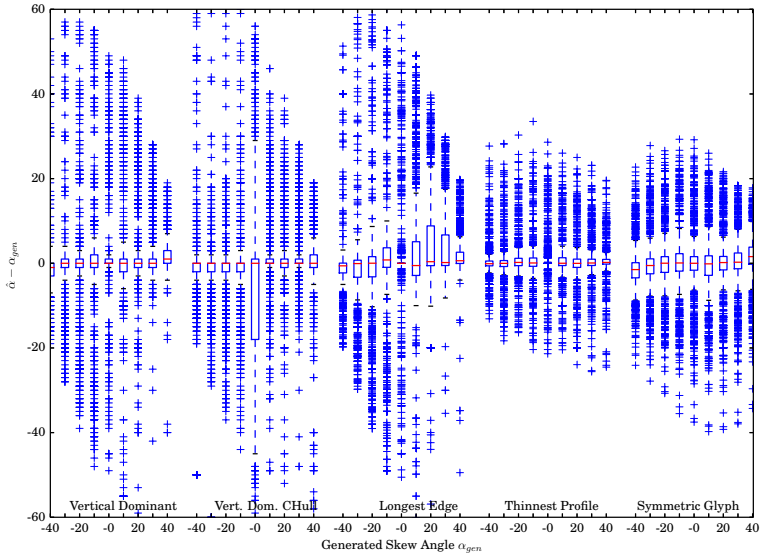

Fig. 11. Skew estimation error as a function of the ground truth skew.

Table 2. Skew estimation accuracy a and standard deviation σ of the combined estimator on synthetic text, dependent on the text length.

	2 characters		3 characters		4 characters	
	a[%]	σ[°]	a[%]	σ[°]	a[%]	σ[°]
VD+VC+SG+TP	75.9	6.4	90.5	3.6	95.4	1.9

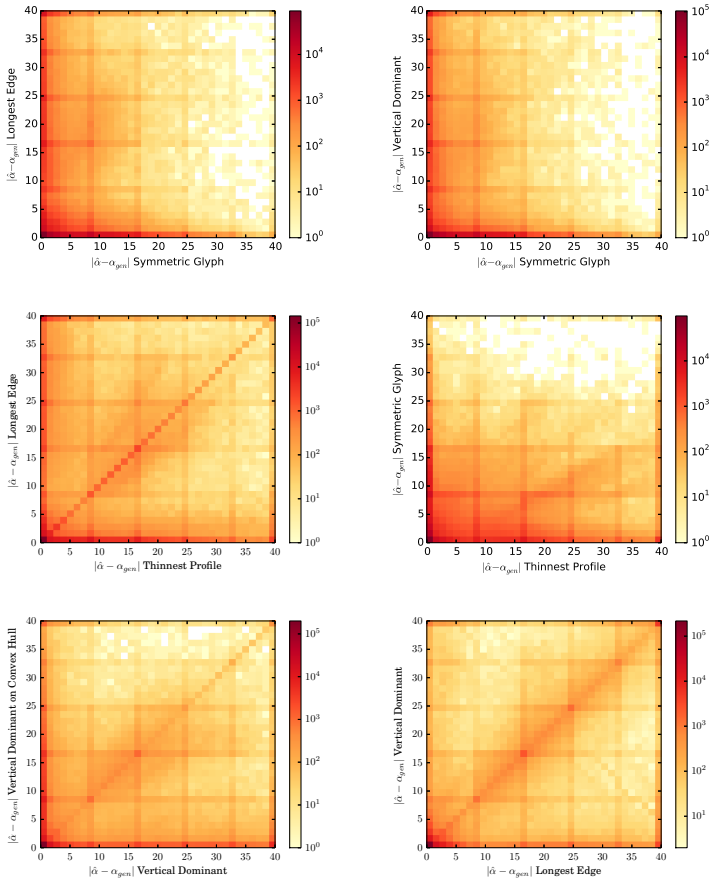

Fig. 12. Skew estimation error $|\hat{\alpha} - \alpha_{\mathrm{gt}}|$ for each estimator pair.

a random skew in the interval $\langle -40°, 40° \rangle$ was applied. The skew estimate is again considered as correct if it differs less than 3° from the ground truth.

In order to benefit from the complementarity of estimators (see Section 5.1), the 4 most complementary estimators (Vertical Dominant, Vertical Dominant on Convex Hull, Thinnest Profile and Symmetric Glyph) were combined into a single estimator. Each estimator votes for its estimated skew $\hat{\alpha}$ with a trained probability, and the resulting skew is obtained as the highest peak in a histogram calculated over all word characters.

It can be observed that the combined estimator predicts skew correctly for 75.9% of words which consist of 2 characters only; the accuracy then grows to 95.4% for 4-character words (see Table 2 for estimation accuracy, Figure 13 for examples of correctly estimated text skew).

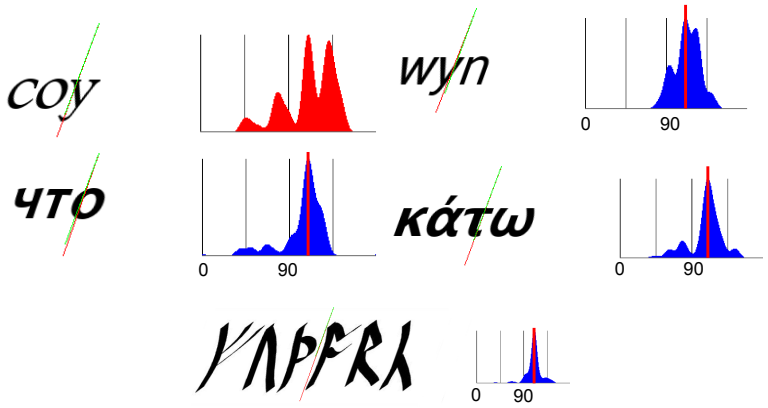

Fig. 13. Samples of a correct skew estimation in the synthetic text dataset. The ground truth skew α_{gt} marked green, the estimated skew $\hat{\alpha}$ marked red.

The estimation typically fails for short words which contain many characters that consist of slanted strokes in an undistorted view (such as `v`, `w` or `y`), which create false peaks in the skew histogram (see Figure 14). However, we can see from the histogram that the confidence of such estimates is relatively low and therefore for practical applications for such low confidence deskewing of the word can be skipped.

5.3 ICDAR 2013 Dataset

In the last experiment, the proposed method was incorporated into the TextSpotter pipeline ([1,2]). In order to evaluate the impact of skew rectification in real-world images, we compared the number of correctly recognized words in the ICDAR 2013 dataset [12] (a word is correctly recognized if all its characters are recognized correctly, using case-sensitive comparison).

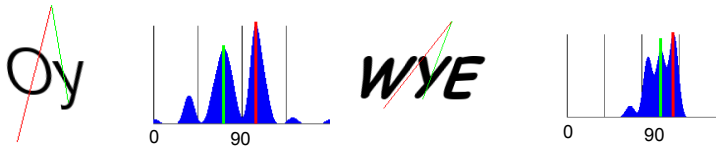

Fig. 14. Samples of an incorrect skew estimation in the synthetic text dataset. The ground truth skew α_{gt} marked green, the estimated skew $\hat{\alpha}$ marked red.

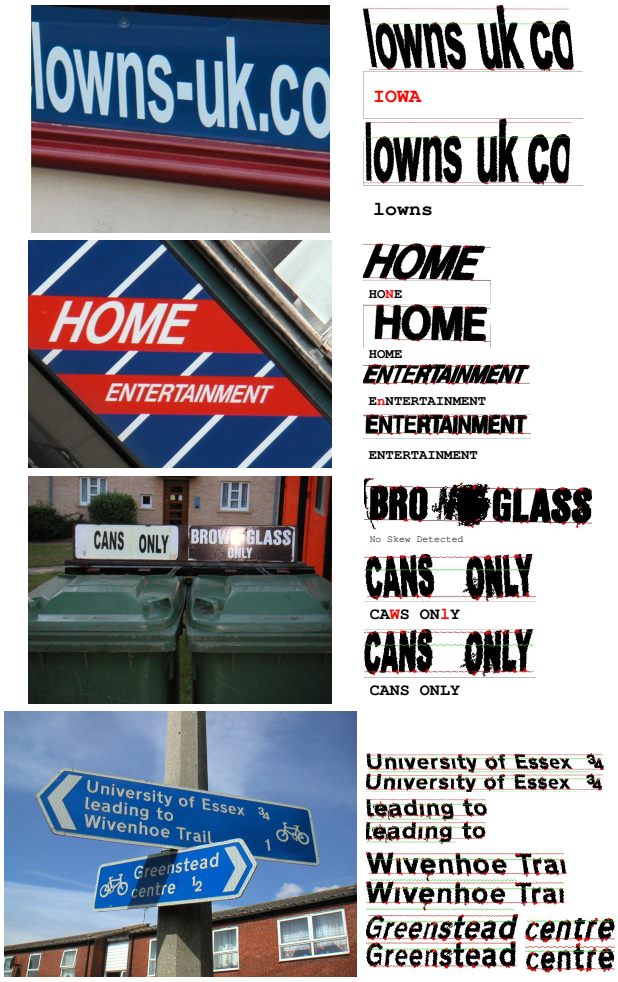

Fig. 15. Samples from the ICDAR 2013 Dataset where the text recognition accuracy was improved. Source image (left), original word segmentation (top right) and rectified word segmentation (bottom right).

Table 3. Text recognition accuracy of the TextSpotter pipeline on the ICDAR 2013 dataset with and without skew rectifcation

	correctly recognized words
Rotation rectification	526 (48.03%)
Rotation and skew rectification	547 (49.95%)
Localization recall	*65.38%*

It can be observed (see Table 3) that the skew rectification using the proposed method improved the recognition accuracy by 21 words, which is relatively significant, given the fact that most of the text in the dataset is horizontal (examples of the words with improved text recognition are shown in the Figure 15). Note that the localization recall of TextSpoter is only 65.38%, therefore the recognition recall of 49.95% implies that 75% of the words that were detected were actually recognized correctly.

6 Conclusions

We have presented five novel character skew estimators, which exploit intuitive glyph properties and which can be efficiently computed in linear time. On synthetically generated data the estimators were able to accurately estimate skew for 5 different scripts in different fonts even for short snippets of text. On real-world images of the ICDAR 2013 dataset the proposed method improved the text recognition accuracy of the state-of-the-art TextSpotter pipeline ([1,2]).

The future work includes learning a custom voting function for the combined estimator and handling of scene text on non-planar surfaces.

Acknowledgment

The authors were supported by the EU project MASELTOV (FP7-ICT-288587) and by the Czech Science Foundation Project GACR P103/12/G084. Lukáš would also like to acknowledge the support of the Google PhD Fellowship in Computer Vision and the Google Research Award.

References

1. Neumann, L., Matas, J.: On combining multiple segmentations in scene text recognition. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, California, US, IEEE (2013) 523–527
2. Neumann, L., Matas, J.: Text localization in real-world images using efficiently pruned exhaustive search. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on, IEEE Computer Society Offices 2001 L Street N.W. Suite 700 Washington, DC 20036-4928, United States, IEEE Computer Society Conference Publishing Services (2011) 687–691

3. Baird, H.S.: Document image analysis. IEEE Computer Society Press, Los Alamitos, CA, USA (1995) 204–208
4. Papandreou, A., Gatos, B.: A novel skew detection technique based on vertical projections. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. (2011) 384–388
5. Epshtein, B.: Determining document skew using inter-line spaces. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. (2011) 27–31
6. Srihari, S., Govindaraju, V.: Analysis of textual images using the hough transform. Machine Vision and Applications **2** (1989) 141–153
7. Bar-Yosef, I., Hagbi, N., Kedem, K., Dinstein, I.: Fast and accurate skew estimation based on distance transform. In: Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on. (2008) 402–407
8. Sun, C., Si, D.: Skew and slant correction for document images using gradient direction. In: Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on. Volume 1. (1997) 142–146 vol.1
9. Ramer, U.: An iterative procedure for the polygonal approximation of plane curves. Computer Graphics and Image Processing **1** (1972) 244 – 256
10. Toussaint, G.: Solving geometric problems with the rotating calipers (1983)
11. Sklansky, J.: Finding the convex hull of a simple polygon. Pattern Recogn. Lett. **1** (1982) 79–83
12. Shahab, A., Shafait, F., Dengel, A.: Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In: Document Analysis and Recognition (ICDAR), 2011 International Conference on. (2011) 1491–1496