Towards Visual Words to Words

Text Detection with a General Bag of Words Representation

Rakesh Mehta
Dept. of Signal Processing,
Tampere Univ. of Technology in Tampere

Ondřej Chum, Jiří Matas
Centre for Machine Peception, Department of Cybernetics,
Faculty of Electrical Engineering, Czech Technical University in Prague

***Abstract*—We address the problem of text localization and retrieval in real world images. We are first to study the retrieval of text images, i.e. the selection of images containing text in large collections at high speed. We propose a novel representation, textual visual words, which describe text by generic visual words that geometrically consistently predict bottom and top lines of text. The visual words are discretized SIFT descriptors of Hessian features. The features may correspond to various structures present in the text - character fragments, individual characters or their arrangements. The textual words representation is invariant to affine transformation of the image and local linear change of intensity. Experiments demonstrate that the proposed method outperforms the state-of-the-art on the MS dataset. The proposed method detects blurry, small font, low contrast, noisy text from real world images.**

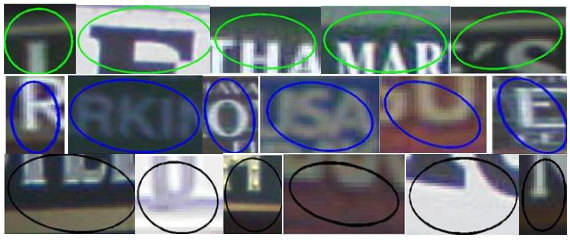

Fig. 1: An example of features whose visual words were learned to predict the position of a text line (from top to bottom): top of the text line, both top and bottom, and bottom.

I. Introduction

Automatically reading text in natural scene images is an open research problem. Detecting (and recognizing) text has many applications: image retrieval, annotation for text-based image search, mobile vision aids, navigation, etc. Reading text from images is commonly broken into main steps. The first – text localization – involves identifying the location of the text in the image, and the second deals with recognizing the words or characters. In this paper we focus on the first step. Despite the effort in the text localization, it remains a challenging open research problem primarily because the images taken in unconstrained setting have large font variations, blurry and low resolution text, noise, small or almost non-readable fonts, occlusion, different orientation, etc.

Current approaches to text localization can be broadly divided into two categories: sliding window methods and Connected Component based. The sliding window approaches [1], [2], [3], [4] scan rectangular regions in the image at different scales. Image regions are classified as text or non-text, based on the features computed from the window. Since rectangular regions are used as processing units, these approaches are robust to pixel level distortions. However, handling geometric transformations like rotation and aspect change is computationally expensive. The state-of-the-art method based on this approach reported processing time of 15 seconds for an 800 $\times$ 1200 image [4].

The connected component based approaches [5], [6], [7], [8] identify the character candidates by filtering out the large background region of image based on certain heuristic property computed at pixel level. Different heuristics have been designed for the character candidate filtering such as Stoke Width Transform (SWT) [5], Color Consistency, Maximal Stable regions [7], [8]. The main advantage of these methods is that the computational complexity is reduced to O(N) because of the filtering step. Its limitations are: (1) heuristics are engineered by assuming certain text property which do not cover the large variations in natural images. For example, color consistency based approaches fail on multi-colored characters and SWT, which assumes constant stoke width fails for non-uniform fonts. (2) Heuristic operations (such as SWT, MSER) are pixels based and hence sensitive to noise, occlusion, etc. These generally involves certain manually tuned parameters which do not work across different dataset. (3) Blurred or low resolution text is difficult to detect because the candidates character regions cannot be identified separately.

We seek to learn the characteristic image patches which appear as parts of text. To learn the patches, an image is represented using Bag-of-Words (BoW) model [9] with Hessian affine region detectors and SIFT descriptor [10]. The Hessian and SIFT features are robust to illumination, blur, affine transformation and scale variations. The number of interest regions detected is higher for Hessian than for the Harris affine and MSER [11], the probability of missing a textual region in first stage is lower.

In the training stage, visual words which correspond to a textual region (textual visual words) are learned. Fig. 1 shows examples of the learned textual regions. In the test images, first the textual regions are identified and based on the geometry of the textual region we estimate the top and bottom line position for textual region. Then, a voting algorithm is proposed to estimate the text lines by aggregating the feature geometry. The proposed approach can be seen as a hybrid of the Region and Connected component based approach. We identify the textual regions (elliptical and not rectangular) and then filter out background region as done in the connected component approach.

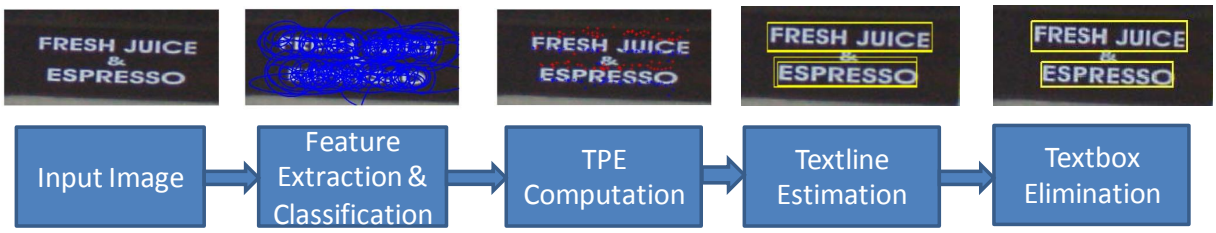

Fig. 2: Detection stage flowchart: Affine covariant features are detected in an image, features described by visual words predicting text lines are selected, text lines are robustly estimated, and text boxes are finally classified based on all content.

The main advantages of the proposed approach are: (1) generality: the proposed method does not assume any specific property of the text, like color or text width, rather it learn the appearance of textual regions from the training images. No parameter tuning is required across different datasets. (2) Robustness to blur, low contrast and noise. (3) Speed: the proposed approach does not use the sliding window to identify the textual regions in the image. The complexity of finding the textual region is O(N). (4) Retrieval application: The BoW representation used in the proposed approach is the state-of-the-art method for the image retrieval system. Therefore, the proposed approach can be seemlessly integrated into a system for retrieval of the text images [12]. (5) Extendable: since the approach is data-driven it can be easily extended to other languages as long as the training data is available for it.

The rest of the paper is organized as follows. In section II, we describe the proposed method for text detection. The experimental setup and the results are discussed in section III. Finally, the paper is concluded in section IV.

II. The Proposed Approach

The main steps of the proposed approach are: (1) extraction of candidate textual regions using Hessian-SIFT features and Textual visual words. (2) Transformation of the textual features into Textlines Point Estimates (TPE). The TPE estimates the relative position of the top and bottom lines w.r.t. feature center. (3) Estimation of the Top and Bottom lines by grouping the TPEs using a RANSAC-based voting algorithm [13]. (4) Verification tests to discard non-text bounding boxes. The structure of the proposed pipelines is presented in Fig. 2. In the training stage we learn the textual visual words which are used in Step (1) and (2) of the proposed pipeline.

A. The Training Stage

For the BoW representation of the training images, first, the Hessian affine-invariant regions are detected and SIFT descriptor are extracted. A feature corresponding to an elliptical region in the image is described by three parameters: scale (s), position (r) and visual word (v). The scale of the feature is defined as the square root of the ellipse area. It is normalized with respect to the bounding box height to make the formulation invariant to the font size. The position parameters is defined as $r = f_y/B_H$, where f_y is the distance of the feature center from the bottom line of the bounding box and B_H is the height of the bounding box. The visual word is assigned by mapping the features descriptor to the nearest entry in the visual vocabulary.

Given the dictionary of the visual words V and the features extracted from the training image, we select a subset of the visual words $T \subset V$ which encodes the textual region in the images and for each textual visual word determine the characteristic scale and position. The characteristic scale defines the size of the text height relative to the size of the features, while the position determines the vertical location of text lines with respect to the feature center. These parameters are used to specify the TPE in the text stage.

A visual word is labeled as textual if it has a high likelihood to be in a text region. The likelihood of the visual word, $v_i \in V$, to be textual is defined as $p_i = t_i/n_i$, where n_i is the count of v_i in the training data and t_i is the count of v_i in a bounding box. The visual words corresponding to the N largest values of p_i are selected as the textual. We chose a large value for N ($N = 60000$), so the visual words which correspond to blurred, low resolution and challenging text are retained.

Based on the position between the text lines, the textual visual words are further categorized as: Top, Bottom and Middle. The idea behind the categorization is illustrated in Fig. 3, where the features close to the top and bottom lines are shown in green and black, the rest are marked blue. It can be observed that the features whose centers are close to top or bottom lines encode upper or lower end of the text respectively. The position of these features is found to be consistent only with respect to one line, therefore, these are used to estimate the location of that particular line. The features in ithe middle of the text line (denoted in blue) are consistent with both top and bottom line and provide a correct estimate for both the lines. The TPE corresponding to the three kind of features are shown in Fig. 3 where it can be observed that the Upper (green) and Lower (black) visual words provide a very precise estimate of the top and bottom location of the text, while the Middle (red and blue) visual words provide a good estimate for both the top and the bottom line.

To find a robust estimate of the characteristic scale and position, the features corresponding to each visual word are clustered together. The features are parameterized using: (1) length of major axis a_x, (2) length of minor axis a_y, (3) vertical position of top line w.r.t feature center $l_t = r/s$, (4) vertical position of bottom line w.r.t feature center $l_b = (r-1)/s$. The first two parameters correspond to the size of the features in X and Y direction, while the next two denote the vertical position of the feature center between text lines. All the parameters are

Fig. 3: Three types of visual words are learned (left): predicting only top of the text line (green), predicting only bottom of the text line (black) and predicting both top and bottom (blue). Feature classification based on location. Each feature provides an estimate of the position of the text line (right). Estimates from features estimating only top (green) or bottom (black) are typically reliable.

normalized by bounding box height to make the formulation invariant to font size. The motivation behind the choice of the parameters is that the features used for estimation of scale and position must be similar in their shape and invariable in their location between text line. However, the invariance of the location can be described with respect to top or bottom or both lines. In order to find weather the features are consistent along the top, bottom or both lines, clustering is performed using three different sets of parameters:

-$C_M \leftarrow Clustering(\mathbf{a_x}, \mathbf{a_y}, \mathbf{l_t}, \mathbf{l_b})$: To group the features whose position is consistent with respect to, both, top and bottom lines (l_t, l_b). The size of the largest cluster is denoted as S_M.

-$C_T \leftarrow Clustering(\mathbf{a_x}, \mathbf{a_y}, \mathbf{l_t}, \mathbf{r})$: To group the features whose position is consistent only with respect to top line (l_t). The size of the largest cluster is denoted as S_T.

-$C_B \leftarrow Clustering(\mathbf{a_x}, \mathbf{a_y}, \mathbf{l_b}, \mathbf{r})$: To group the features whose position is consistent only with respect to bottom line (l_b). The size of the largest cluster is denoted as S_B.

The vectors $\mathbf{a_x}, \mathbf{a_y}, \mathbf{l_b}, \mathbf{l_t}, \mathbf{r}$ represent the parameters values for the feature corresponding to a textual visual word. Mean-Shift is used for clustering the features as it gives the locations of the maxima of the feature density. The centroid of the largest feature cluster is used for estimating the values of the scale and position for the visual word. Given the l_t and l_b values, finding the characteristic scale and position is trivial. Three clustering scenarios C_M, C_T, C_B represent the characteristic scale and position estimated under different consistency definition. The large cluster reflects better consistency of the features and is considered for classification. The visual word is classified as top, bottom or middle based on the position r estimated from cluster consistent with both lines. It is categorized as bottom if $r < 0.1$ and $C_B > 2 \cdot C_M$, as top if $r > 0.9$ and $C_B > 2 \cdot C_M$, and middle otherwise.

B. Generating training data:

Training data play a crucial role in the performance of the text detector as we do not assume any specific property of the text. The training data must be large to capture the real world variations in text. Currently available text training datasets (ICDAR, SWT, etc) do not fulfill these requirements, as these are too small (of the order of few hundreds) and do not capture the real world variations of the text. In order to overcome these limitation we generate the training data using real world images. We use the Oxford 100k image data [14] and find the text bounding boxes using the TextSpotter text localization system [15]. The Oxford 100K dataset consists of images collected from Flicker in uncontrolled environment, thus, it contain the challenging text examples. TextSpotter is a publicly available text detection system which has achieved state-of-the-art on real world images.

C. Testing Stage

1) Textual Regions Detection: A test image is described by the BoW. The regions which correspond to the textual visual words are identified as textual regions. The scale for the textual regions is computed. This can be interpreted as a filtering step where the candidate text regions are identified and retained and the background is discarded.

2) Computation of Textline Point Estimates (TPE): The textual regions are transformed into the Textlines Points Estimates. Given a textual feature in the test image at location (x, y), TPE estimates the points (x, y_t) and (x, y_b) which are likely to pass through the top and the bottom line respectively. If the textual region correspond to the textual visual word $v_i \in T$ with characteristic scale s_i and position r_i, then the TPE Y-coordinates are given as,

$$y_t = y + \frac{r_i \times S}{s_i}, \quad y_b = y + \frac{(r_i - 1) \times S}{s_i} \tag{1}$$

where S is the scale of the feature in the text image.

In the proposed TPE formulation we use the top and bottom line coordinates for each feature because these parameters maintain consistency in training images (as observed by the large cluster size in training stage).

3) Textline estimation using Voting: A voting algorithm is used for estimation of the text line from the TPEs. It is based on NAPSAC [16] and LO-RANSAC [13] with iterative re-sampling and some modification for our specific application. The detailed algorithm is presented in Algo. 1. As in RANSAC we start by selecting a minimum number of samples for generating the model. The first sample is randomly drawn and the second is chosen from a rectangular region center around the first one. The model of top and bottom line is generated from these samples. For local optimization of the model we use the following function:

$LO\_Joint(I)$: Iterative re-sampling is performed by selecting the samples from inliers set I and the following cost function is maximized

$$f_{joint} = k \cdot (n_T + n_B) + n_M, \tag{2}$$

where n_T, n_B and n_M are number of inliers which belong to the Top, Bottom and Middle textual visual words. The inliers are the samples which are consistent with both the lines. The joint optimization enforces the line to be close to parallel and provides robustness to the model as all the samples are considered in the cost function.

Finally the model is selected if its overlap with selected models is less than a threshold or the joint cost function (Eq 2) is more than those.

Algorithm 1 Textline estimation.

for $j = 1$ to J **do**
  Randomly draw two TPE
  $[L] \leftarrow$ Estimate line pair joining the TPE
  $I_1 \leftarrow$ find_inliers(L, θ)
  $[L^{LSq}] \leftarrow$ Least square line estimate using I_1
  $I_2 \leftarrow$ find_inliers$(L^{LSq}, 0.5 \cdot \theta)$
  $L^j = LO\_Joint(I_2)$
  Check overlap and cost function.
end for

TABLE I: Performance comparison on the MS dataset.

Method	Precision	Recall	$f-measure$
SWT	42	54	47
TextSpotter	50	21	30
Phan et. al. method [17]	51	50	51
Yin et. al. method [8]	41	66	51
Proposed	46	60	52

4) Determining the Bounding Box Width: The text is assumed to be aligned in a near horizontal direction. The width of the text box is determined by grouping the TPEs consistent with the selected model. First, we find the *connected group* of the TPEs along the X direction. A *connected group* is a cluster in which the distance between each TPE and its neighbor is always less than a threshold $\phi = w_1 + w_2 \cdot h$, where h is the average height of the randomly drawn TPE. It gives a higher threshold to the large TPE because for larger fonts it is more difficult to have robust estimates of both the lines. The constant term for the threshold is fixed relative to the image height $w_1 = ImageHeight/30$ and the TPE dependent term w_2 is set to 0.3. To deal with the occlusion and word gaps in text line, we join the adjacent connected groups if they have similar density and a minimum number of TPE in it.

5) Discarding the Non-Text Bounding Boxes: To filter out false components we apply a fairly flexible set of rules on the models generated by above mentioned algorithm. The textual regions have high density of textual visual words, therefore for each text box we compute the ratio of the number of textual visual words to the total number of textual visual words in it. A box is discarded if the ratio is less than a particular threshold. It helps in identifying the false positives in highly textured regions such as grass, or tree leaves. Repetitive structures such as windows, bricks, railing result in a number of false bounding boxes. The regions corresponding to these structure have high frequency of certain visual words. We find the proportion of the unique textual visual words for each bounding box and reject those with low proportion. Each text lines pair should have a minimum number of inliers to support it, we reject all the models which have less than 5 TPE consistent with it. The slope of the top and the bottom lines are also compared and model is rejected if the difference between the slopes is more than 45^o.

III. Experiments

The performance of the algorithm is evaluated on two different task: (1) Text detection and (2) Text image retrieval.

Fig. 4: Text detection results on several images from MS and ICDAR dataset.

Detection performance is evaluated on two standard widely-use public text image datasets: MS and ICDAR 2013. The MS dataset [5] consists of 307 images taken from Streetview. It contains a number of repetitive pattern and low contrast text. The ICDAR 2013 Robust Reading Competition dataset [18] consists of 235 test images and is most widely used dataset for evaluation of text detection. The MS dataset is textline annotated, while the ICDAR dataset is word annotated. The proposed approach is for textline detection. Nevertheless, the performance is evaluated on the word annotated ICDAR since it is most widely used dataset for text detection. For a fair comparison with earlier work [5], [8], [17] we follow the 2003 ICDAR protocol [19] for MS dataset and 2013 ICDAR protocol for the ICDAR dataset. Text image retrieval experiment was performed on a 5 million Flicker images database.

A. Implementation details

In the training stage, we used the visual vocabulary of size 10^6 with the 100K Oxford dataset. The number of textual visual word was set to 60,000. For the voting algorithm, the number of iteration J was set to the number of textual visual words identified in the image. The number of iteration for the re-sampling step (LO_Joint) was set to 5. The threshold value for inliers is $\theta = 0.6 \times h$, where h is the height of the randomly selected TPE. The value of k for the cost function was 2. It was observed that the performance does not vary for values up to 5. Finally, the threshold for filtering out the bounding boxes in the last stage was tuned using the training images of the datasets.

B. Text Detection

The performance of the method on the MS dataset is summarized in Table I. The results are compared with the SWT [5], TextSpotter [7], MSER based method [8], and Boosting based approach [17] and the proposed method achieves the highest f-measure. Good results were obtained even in images are taken in challenging scenarios, such as low contrast, blur, noise. Examples of the detected text are shown in Fig. 5.

On ICDAR-2013 dataset the performance of the algorithm is: recall: 58.35, precision: 68.58, f-measure 62.99. The performance of the ICDAR robust reading competition winner is: 69.28, 88.80, 77.83. There are two main reasons behind the significant change in the performance of the proposed method. First, the ICDAR dataset is word annotated, while the output of the proposed algorithm are the bounding boxes which correspond to the textline. The second reason is that the proposed method does not work well when there are less than two words in a text line or the words are isolated. The voting algorithm requires a certain number of TPEs aligned in linear

Fig. 5: The algorithm detects text from blurry and low contrast images. Examples are from the MS dataset.

Fig. 6: Sample images with ranks retrieved using a query composed of the top 5000 Textual Visual Words.

manner to generate textlines. Both isolated words and textlines with less than two words are not rare in the ICDAR dataset. Example test images from ICDAR and MS datasets are shown in Fig. 4.

C. Text Retrieval

To test the algorithm for retrieval of images depicting text, we use a dataset of over 5 million images downloaded from Flickr. The dataset contain both text and non-text images. The dataset images are represented by BoW and efficient query evaluation is implemented through an inverted file. The retrieval was tested by issuing a query comprising the top 5000 visual words learned to represent text (*i.e.* with the highest textual prior p_i). The images are ranked using the *tf-idf* scoring. A uniformly sampled set from the top 50000 retrieved images was visually analyzed and it was observed that all the images contain text. As expected the high ranked images contain large textual regions, see Fig. 6. The time needed to retrieve 100000 images from the indexed database is less than 1 second.

To analyze the 'textuality' of the retrieved image we found the average number of textlines detected per image. The results are shown in Fig. 7. The average number of textlines per image decrease with the rank.

IV. Conclusion

A novel approach for text detection and text image retrieval from natural images has been presented. The key contribution is the novel image representation, Textual Visual words, which are based on the BoW model. We also propose a voting based algorithm for the textline estimation from the extracted features. Experiments demonstrate that the proposed approach can detect text in challenging scenarios and can retrieve highly textual images from the large images datasets at a very high speed.

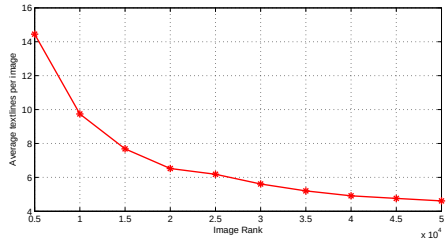

Fig. 7: The average number of text lines detected in the retrieved images as a function of the rank of the image.

Acknowledgment

Rakesh Mehta was supported by Tampere Graduate School in Information Science and Engineering (TISE). Jiří Matas was supported by EC project FP7-ICT-288587 MASELTOV and by the Technology Agency of the Czech Republic research program TE01020415 (V3C-Visual Computing Competence Center). Ondřej Chum was supported by MSMT LL1303 ERC-CZ grant.

References

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

[2] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. L. Yuille, and C. Koch, "Adaboost for text detection in natural scene." in *ICDAR*, 2011, pp. 429–434.

[3] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE TCSVT*,, vol. 12, no. 4, pp. 256–268, 2002.

[4] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *ICCV*, 2011, pp. 1457–1464.

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

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

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

[8] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE TPAMI*, vol. 36, no. 5, pp. 970–983, 2014.

[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*. IEEE, 2003, pp. 1470–1477.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1-2, pp. 43–72, 2005.

[12] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*. Springer, 2008, pp. 304–317.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*. IEEE, 2007, pp. 1–8.

[15] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *IEEE CVPR*. IEEE, 2012, pp. 3538–3545.

[16] D. Myatt, P. Torr, J. Bishop, R. Craddock, and S. Nasuto, "Napsac: High noise, high dimensional robust estimation - it's in the bag," in *BMVC*, vol. 2, 2002, pp. 458–467.

[17] T. Q. Phan, P. Shivakumara, and C. L. Tan, "Detecting text in the real world," in *ACM ICM*. ACM, 2012, pp. 765–768.

[18] D. Karatzas *et al.*, "Icdar 2013 robust reading competition," in *ICDAR*. IEEE, 2013, pp. 1484–1493.

[19] S. Lucas *et al.*, "Icdar 2003 robust reading competitions," in *ICDAR*, vol. 2. IEEE, 2003, pp. 682–682.