ROBUST TEXT DETECTION IN NATURAL IMAGES WITH EDGE-ENHANCED MAXIMALLY STABLE EXTREMAL REGIONS

Huizhong Chen[1], *Sam S. Tsai*[1], *Georg Schroth*[2], *David M. Chen*[1], *Radek Grzeszczuk*[3] *and Bernd Girod*[1]

[1]Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA
[2]Institute of Media Technology, Technische Universität München, Germany
[3]Nokia Research Center, Palo Alto, CA 94304, USA

ABSTRACT

Detecting text in natural images is an important prerequisite. In this paper, we propose a novel text detection algorithm, which employs edge-enhanced Maximally Stable Extremal Regions as basic letter candidates. These candidates are then filtered using geometric and stroke width information to exclude non-text objects. Letters are paired to identify text lines, which are subsequently separated into words. We evaluate our system using the ICDAR competition dataset and our mobile document database. The experimental results demonstrate the excellent performance of the proposed method.

Index Terms— Text detection, maximally stable extremal regions, connected component analysis

1. INTRODUCTION

Mobile visual search has gained popular interest with the increasing availability of high-performance, low-cost camera-phones. In recent years, visual search systems have been developed for applications such as product recognition [1, 2] and landmark recognition [3]. In these systems, local image features [4, 5, 6] are extracted from images taken with a camera-phone and are matched to a large database using visual word indexing techniques [7, 8]. Although current visual search technologies have reached a certain level of maturity, they have largely ignored a class of informative features often observed in images: text. In fact, text is particularly interesting because it provides contextual clues for the object appearing inside an image. Given the vast number of text-based search engines, retrieving an image using the embedded text offers an efficient supplement to the visual search systems.

As an essential prerequisite for text-based image search, text within images has to be robustly located. However, this is a challenging task due to the wide variety of text appearances, such as variations in font and style, geometric and photometric distortions, partial occlusions, and different lighting conditions. Text detection has been considered in many recent studies and numerous methods are reported in the literature [9, 10, 11, 12, 13, 14, 15, 16, 17]. These techniques can be classified into two categories: texture-based and connected component (CC)-based.

Texture-based approaches view text as a special texture that is distinguishable from the background. Typically, features are extracted over a certain region and a classifier (trained using machine learning techniques or by heuristics) is employed to identify the existence of text. In [11], Zhong et al. assume text has certain horizontal and vertical frequencies and extract features to perform text detection in the discrete cosine transform domain. Ye et al. collect features from wavelet coefficients and classify text lines using SVM [12]. Chen et al. feed a set of weak classifiers to the Adaboost algorithm to train a strong text classifier [13, 14].

(a) Detected MSER (b) Text candidates (c) Detected text

Fig. 1. Extracting text from a natural image. (a): Detected MSER for dark objects on bright background. (b): After geometric and stroke width filtering, text candidates are pairwise grouped to form text lines. The text lines are shown by the red lines. (c): Text line verification rejects false positives and the detected text is highlighted by the blue box.

As opposed to texture-based methods, the CC-based approach extracts regions from the image and uses geometric constraints to rule out non-text candidates. The top scoring contestant in [15] applies an adaptive binarization method to find CCs. Text lines are then formed by linking the CCs based on geometric properties. Recently, Epshtein et al. [16] proposed using the CCs in a stroke width transformed image, which is generated by shooting rays from edge pixels along the gradient direction. Shivakumara et al. extract CCs by performing K-means clustering in the Fourier-Laplacian domain, and eliminate false positives by using text straightness and edge density [17].

In this work, we propose a novel CC-based text detection algorithm, which employs Maximally Stable Extremal Regions (MSER) [18] as our basic letter candidates. Despite their favorable properties, MSER have been reported to be sensitive to image blur [19]. To allow for detecting small letters in images of limited resolution, the complimentary properties of Canny edges and MSER are combined in our edge-enhanced MSER. Further we propose to generate the stroke width transform image of these regions using the distance transform to efficiently obtain more reliable results. The geometric as well as stroke width information are then applied to perform filtering and pairing of CCs. Finally, letters are clustered into lines and additional checks are performed to eliminate false positives. The overall process of the text detection is illustrated in Fig. 1. In comparison to previous text detection approaches, our algorithm offers the following major advantages. First, the edge-enhanced MSER detected in the query image can be used to extract feature descriptors like [4, 5] for visual search. Hence our text detection can be combined with visual search systems without further computational load to detect interest regions. Further, our system provides a reliable binarization for the detected text, which can be passed to OCR for

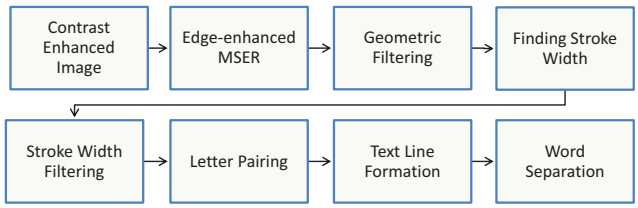


Fig. 2. System flowchart

text recognition. Finally, the proposed algorithm is simple and efficient. MSER as well as the distance transform can be very efficiently computed [20, 21] and determining the stroke width only requires a lookup table (Section 2.3).

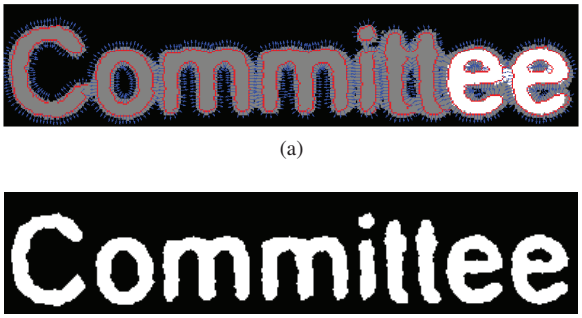
The remainder of this paper is organized as follows. In Section 2, we describe the individual steps of our text detection algorithm. Section 3 demonstrates the robust performance of the proposed system and Section 4 concludes the paper.

2. THE TEXT DETECTION ALGORITHM

The flowchart of our text detection algorithm is shown in Fig.2. At the input of the system, the image intensities are linearly adjusted to enhance the contrast. Subsequently, MSER regions are efficiently extracted from the image [20] and enhanced using Canny edges obtained from the original gray-scale image (Section 2.1). As a next step, the resulting CCs are filtered using geometric constraints on properties like aspect ratio and number of holes (Section 2.2). The stroke width information is robustly computed using a distance transform (Section 2.3) and objects with high variation in stroke width are rejected. Text candidates are grouped pairwise and form text lines. Finally, words within a text line are separated, giving segmented word patches at the output of our system.

2.1. Edge-enhanced MSER

As the intensity contrast of text to its background is typically significant and a uniform intensity or color within every letter can be assumed, MSER is a natural choice for text detection. While MSER has been identified as one of the best region detectors [19] due to its robustness against view point, scale, and lighting changes, it is sensitive to image blur. Thus, small letters cannot be detected or distinguished in case of motion or defocus blur by applying plain MSER to images of limited resolution. Fig. 3a shows an example where multiple letters are identified as a single MSER region. To cope with blurred images we propose to combine the complimentary properties of Canny edges [22] and MSER. The outline of extremal regions can be enhanced by applying the precisely located but not necessarily connected Canny edges. As shown in Fig.3a, we remove the MSER pixels outside the boundary formed by the Canny edges. This is achieved by pruning the MSER along the gradient directions (indicated by the blue arrows) computed from the original gray-scale image. Since the type of the letter (bright or dark) is known during the MSER detection stage, the gradient directions can be adapted to guarantee that they point towards the background. Fig.3b shows the edge-enhanced MSER, which provides a significantly improved representation of the text where individual letters are separated. This not only improves the performance of geometric filtering (Section 2.2), but also increases the repeatability of MSER based feature matching under different image blur conditions.



(a)

(b)

Fig. 3. Edge-enhanced MSER. (a) Detected MSER for blurred text. Canny edges are shown in red lines and the blue arrows indicate gradient directions. (b) MSER after pruning along the gradient.

(a) Distance map (b) stroke width image

Fig. 4. Finding the stroke width information. (a) The distance transformed image. (b) Stroke width image is formed by propagating the stroke width information from the ridge to the boundary. The numbers label half of the stroke width.

2.2. Geometric Filtering

With the extraction of edge-enhanced MSER, we obtain a binary image where the foreground CCs are considered as letter candidates. As in most state-of-the-art text detection systems, we perform a set of simple and flexible geometric checks on each CCs to filter out non-text objects. First of all, very large and very small objects are rejected. Then, since most letters have aspect ratio being close to 1, we reject CCs with very large and very small aspect ratio. A conservative threshold on the aspect ratio is selected to make sure that some elongated letters such as 'i' and 'l' are not discarded. Lastly, we eliminate objects which contain a large number of holes, because CCs with many holes are unlikely to be letter candidates.

2.3. Finding Stroke Width by Distance Transform

The importance of stroke width information has been emphasized in several recent studies [23, 24, 16]. Motivated by Epshtein's work on the Stroke Width Transform (SWT) [16], we develop an image operator to transform the binary image into its stroke width image. The stroke width image is of the same resolution as the original image, with the stroke width value labeled for every pixel. We determine the stroke width using a novel approach based on the distance transform, which differs drastically from the SWT proposed in [16]. Epshtein's SWT forms CCs by shooting rays from the edge pixels along the gradient, and only keeps the rays if they are terminated by another edge pixel having the opposite gradient direction. This method does not work well when the opposite stroke edges are not parallel. Consequently, the stroke width CCs formed by the SWT often have undesirable holes appearing in curved strokes or stroke joints. In contrast to the SWT, our proposed method guarantees that

the SW information is provided at every pixel of the original CC with any stroke shape. In our algorithm, the Euclidean distance transform is applied to label each foreground pixel with the distance to its nearest background pixel. As can be seen in Fig.4a, the ridge values of the distance map correspond to half the width of the stroke. Then, we propagate the stroke width information from the ridge to the boundary of the object, along the "downhill" direction. The stroke width image is shown in Fig.4b. Our method bypasses the need to locate ridge pixels by iteratively propagating the stroke width information, starting from the maximum value to the minimum value of the distance map. The procedure is outlined in Algorithm 1.

Algorithm 1 Finding stroke width

Input: binary image BW
Output: stroke width image SW
  D := DistanceTransform(BW);
  D := round(D);
  for p = each foreground pixel in D **do**
    pVal := D(p);
    LookUp(p) := p's 8 neighbors whose value $<$ pVal;
  end for
  {LookUp can be efficiently computed without FOR loop.}
  MaxStroke := max(D);
  for Stroke = MaxStroke **to** 1 **do**
    StrokeIndex := find(D==Stroke);
    NeighborIndex = Lookup(StrokeIndex);
    while NeighborIndex not empty **do**
      D(NeighborIndex) := Stroke;
      NeighborIndex := Lookup(NeighborIndex);
    end while
  end for
  return SW := D;

The output of the Algorithm 1 is an image where each pixel is assigned a value equal to half of the stroke width. Assuming that the stroke width of characters has a low variation, we exclude CCs with a large standard deviation. The rejection criterion is $std/mean > 0.5$, which is invariant to scale changes. This threshold was obtained from the training set of the ICDAR competition database.

2.4. Text Line Formation and Word Separation

Text lines are important cues for the existence of text, as text almost always appear in the form of straight lines or slight curves. To detect these lines, we first pairwise group the letter candidates using the following rules. As letters belonging to the same text line are assumed to have similar stroke width and character height, two letter candidates are paired if the ratio of their stroke width medians is lower than 1.5 and their height ratio is lower than 2 (taking upper and lower case letters into account). Additionally, two CCs should not be paired if they are very distant.

Subsequently, text lines are formed based on clusters of pairwise connected letter candidates. A straight line is fitted to the centroids of pairs of letter candidates within each cluster and the line that intersects with the largest number of text candidates is accepted. The process is iterated until all text candidates have been assigned to a line, or if there are less than three candidates available within the cluster. A line is declared to be a text line if it contains three or more text objects.

We filter out improbable text lines by two additional validation steps. As shown in Fig. 1b, a false text line is formed along the repetitive windows. Repeating structures such as windows and bricks are commonly seen in urban images, resulting a large number of false positives. This can be avoided by applying template matching among the letter candidates. A text line is rejected if a significant portion of the objects are repetitive. Also, based on the observation that most letters have low solidity (proportion of the object pixels in the convex hull), a text line is rejected if most of the objects within that line have a very large solidity.

As a final step, text lines are split into individual words by classifying the inter letter distances into two classes: the character spacings and the word spacings. We calculate the distance between the vertical projections of each character along the text line and perform a two class classification using the Otsu's method [25].

3. EXPERIMENTAL RESULTS

To evaluate our text detection algorithm, we apply it to two different test sets. As a primary test we use the well-known ICDAR text detection competition data set [26, 15], which was also used as a benchmark for [16, 27, 28]. Further, we apply our algorithm to a document database, which we have created to test a document retrieval system based on text as well as low bit rate features in [29]. The results are shown in the following sections.

3.1. ICDAR Text Detection

Two competitions (ICDAR 2003 and 2005) have been held to evaluate the performance of various text detection algorithms [26, 15]. To validate the performance of our proposed system, we use the metrics defined in [15] and run our algorithm on the ICDAR competition dataset. The precision and recall are defined as $p = \sum_{r_e \in E} m(r_e, T)/|E|$ and $r = \sum_{r_t \in T} m(r_t, E)/|T|$, where $m(r, R)$ is the best match for a rectangle r in a set of rectangles R, E and T are our estimated rectangles and the ground truth rectangles respectively. An f metric is used to combine the precision and recall into one single measure: $f = 1/(\alpha/p + \alpha/r)$, where $\alpha = 0.5$ gives equal weights to precision and recall. Since it is unlikely to produce estimated rectangles which exactly align with the manually labeled ground truth, the f metric can vary from $0.8 - 1.0$ even when all text is correctly localized.

We show the text detection performance on the dataset in Table 1. The results in the lower half include the contestants in [26, 15], where Hinnick Becker's approach achieves the highest f score of 0.62. The upper half contains the results of our text detection system and the state-of-the-art algorithms. Our algorithm achieves an f score similar to Epshtein [16], outperforming all results from the text detection competition. The complexity of our overall detection system is mainly driven by the MSER extraction stage, which requires less than 200 ms for an image resolution of 1280x960 on a 2.5 GHz CPU.

3.2. Document Title Text Detection

As a second test we apply the proposed text detection system to perform a mobile paper search by recognizing the document title in images recorded with a camera-phone and querying databases like Google Scholar in [29]. The first step for such a mobile paper search system is to detect the title text within the document images as shown in Fig. 5. The performance of our text detection algorithm is evaluated by checking the correctly detected bounding boxes around the title text. We use a stringent criterion and declare a title to be correctly detected only when all letters within the title are detected. Out

Table 1. Evaluation of text detection algorithms.

Algorithm	precision	recall	f
Our system	**0.73**	0.60	**0.66**
Epshtein [16]	**0.73**	0.60	**0.66**
Minetto [27]	0.63	0.61	0.61
Fabrizio [28]	0.46	0.39	0.43
Hinnerk Becker	0.62	**0.67**	0.62
Alex Chen	0.60	0.60	0.58
Ashida	0.55	0.46	0.50
HWDavid	0.44	0.46	0.45
Wolf	0.30	0.44	0.35
Qiang Zhu	0.33	0.40	0.33
Jisoo Kim	0.22	0.28	0.22
Nobuo Ezaki	0.18	0.36	0.22
Todoran	0.19	0.18	0.18
Full	0.10	0.06	0.08

of 501 SVGA size images, we are able to correctly identify 477 titles, achieving a performance score of 95%. The cases where the detection fails are due to excessive blur and out of focus.

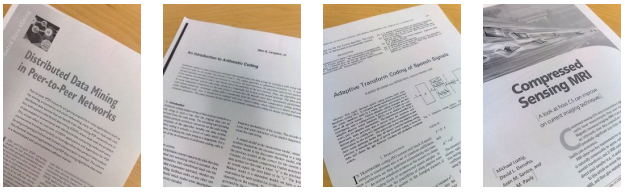


Fig. 5. Document images under various viewpoints.

4. CONCLUSION

In this work, a novel text detection algorithm is proposed, which employs Maximally Stable Extremal regions as basic letter candidates. To overcome the sensitivity of MSER with respect to image blur and to detect even very small letters, we developed an edge-enhanced MSER which exploits the complimentary properties of MSER and Canny edges. Further, we present a novel image operator to accurately determine the stroke width of binary CCs. Our proposed method has demonstrated state-of-the-art performance for localizing text in natural images. The detected text are binarized letter patches, which can be directly used for text recognition purposes. Additionally, our system can be efficiently combined with visual search systems by sharing MSER as interest regions.

5. ACKNOWLEDGEMENT

The authors would like to thank Gabriel Takacs, Vijay Chandrasekhar and Ngai-Man Cheung at Electrical Engineering Department, Stanford University, for their valuable feedback and suggestions. We also sincerely thank Ramakrishna Vedantham at Nokia Research, Palo Alto, for insightful discussions.

6. REFERENCES

[1] S. S. Tsai, D. Chen, V. Chandrasekhar, G. Takacs, N. M. Cheung, R. Vedantham, R. Grzeszczuk, and B. Girod, "Mobile product recognition," in *Proc. ACM Multimedia 2010*, 2010.

[2] D. Chen, S. S. Tsai, C. H. Hsu, K. Kim, J. P. Singh, and B. Girod, "Building book inventories using smartphones," in *Proc. ACM Multimedia*, 2010.

[3] G. Takacs, Y. Xiong, R. Grzeszczuk, V. Chandrasekhar, W. Chen, L. Pulli, N. Gelfand, T. Bismpigiannis, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proc. ACM Multimedia Information Retrieval*, 2008, pp. 427–434.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346 – 359, 2008.

[6] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients. a low bit-rate feature descriptor," in *CVPR*, 2009, pp. 2504 –2511.

[7] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.

[8] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Inverted Index Compression for Scalable Image Matching," in *Proc. of IEEE Data Compression Conference (DCC)*, Snowbird, Utah, March 2010.

[9] J. Liang, D. Doermann, and H. P. Li, "Camera-based analysis of text and documents: a survey," *IJDAR*, vol. 7, no. 2-3, pp. 84–104, 2005.

[10] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977 – 997, 2004.

[11] Y. Zhong, H. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 385 –392, 2000.

[12] Q. Ye, Q. Huang, W. Gao, and D. Zhao, "Fast and robust text detection in images and video frames," *Image Vision Comput.*, vol. 23, pp. 565–576, 2005.

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

[14] X. Chen and A. L. Yuille, "A time-efficient cascade for real-time object detection: With applications for the visually impaired," in *CVPR - Workshops*, 2005, p. 28.

[15] S. M. Lucas, "ICDAR 2005 text locating competition results," in *ICDAR*, 2005, pp. 80 – 84 Vol. 1.

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

[17] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A laplacian approach to multi-oriented text detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 412 –419, feb. 2011.

[18] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British Machine Vision Conference*, 2002, vol. 1, pp. 384–393.

[19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *Int. J. Comput. Vision*, vol. 65, pp. 43–72, 2005.

[20] D. Nistér and H. Stewénius, "Linear time maximally stable extremal regions," in *ECCV*, 2008, pp. 183–196.

[21] D. G. Bailey, "An efficient euclidean distance transform," in *Combinatorial Image Analysis, IWCIA*, 2004, pp. 394–408.

[22] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, pp. 679–698, 1986.

[23] A. Srivastav and J. Kumar, "Text detection in scene images using stroke width and nearest-neighbor constraints," in *TENCON 2008 - 2008 IEEE Region 10 Conference*, 2008, pp. 1–5.

[24] K. Subramanian, P. Natarajan, M. Decerbo, and D. Castanon, "Character-stroke detection for text-localization and extraction," in *ICDAR*, 2007, vol. 1, pp. 33–37.

[25] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62 –66, 1979.

[26] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R Young, "ICDAR 2003 robust reading competitions," in *ICDAR*, 2003, vol. 2, p. 682.

[27] R. Minetto, N. Thome, M. Cord, J. Fabrizio, and B. Marcotegui, "Snoopertext: A multiresolution system for text detection in complex visual scenes.," in *ICIP*, 2010, pp. 3861–3864.

[28] J. Fabrizio, M. Cord, and B. Marcotegui, "Text extraction from street level images," in *CMRT*, 2009, pp. 199–204.

[29] S. S. Tsai, H. Chen, D. M. Chen, G. Schroth, R. Grzeszczuk, and B. Girod, "Mobile visual search on papers using text and low bit-rate features," in *ICIP*, 2011.