Text Localization and Recognition in Complex Scenes Using Local Features

Qi Zheng[1], Kai Chen[1], Yi Zhou[1], Congcong Gu[1], and Haibing Guan[2]

[1] School of Information Security Engineering, Shanghai Jiao Tong University
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University

Abstract. We describe an approach using local features to resolve problems in text localization and recognition in complex scenes. Low image quality, complex background and variations of text make these problems challenging. Our approach includes the following stages: (1) Template images are generated automatically; (2) SIFT features are extracted and matched to template images; (3) Multiple single-character-areas are located using segmentation algorithm based upon multiple-size sliding subwindows; (4) An voting and geometric verification algorithm is used to identify final results. This framework thus is essentially simple by skipping many steps, such as normalization, binarization and OCR, which are required in previous methods. Moreover, this framework is robust as only SIFT feature is used. We evaluated our method using 200,000+ images in 3 scripts (Chinese, Japanese and Korean). We obtained average single-character success rate of 77.3% (highest 94.1%), average multiple-character success rate of 63.9% (highest 89.6%).

1 Introduction

Our goal is to read text from an image in complex scenes. There are many applications for such a technology, for example, recognizing sign from natural scenes, recognizing book/CD cover, license plate recognition, image and video search engine and web mining.

However, variations of text due to differences in size, style, orientation, and alignment, as well as low image quality, complex background and deformation in complex scenes make text localization and recognition a challenging task.

Previous methods [6–9] often consist of following stages, as shown in Figure 1(a), (1) Text localization and extraction; (2) Preprocessing; (3) OCR recognition. Of note, every stage consists of multiple steps that each has its own algorithm and usually operates sequentially.

Local features [1–5], which are distinctive and robust to noise, complicated background, and many kinds of geometric and photometric deformations, have been applied successfully in a wide range of systems and applications, such as wide baseline matching, object recognition, image retrieval, building panoramas, and video data mining. Moreover, as Figure 2 shows, local features matching can be potentially extended to text recognition problems.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 121–132, 2011.

Inspired by the success of utilizing local features in image matching, we describe a local-feature-based approach for text localization and recognition. Considering the difference between text recognition and general image matching, we improve several steps in our approach accordingly, (1) we develop a new template-build method to automatically generate template images while eliminate the influence of complex scenes; (2) we develop a voting algorithm and a geometric verification algorithm for optimizing matching results and locating text; (3) we develop a segmentation algorithm based upon multiple-size sliding sub-windows to handle multiple characters efficiently.

Our framework, as shown in Figure 1(b), is essentially simple by skipping many usual steps, such as normalization, binarization, layout analysis and OCR, which are required in OCR-based methods. Moreover, this framework is robust and applicable in complex scenes as only SIFT feature is used during the process.

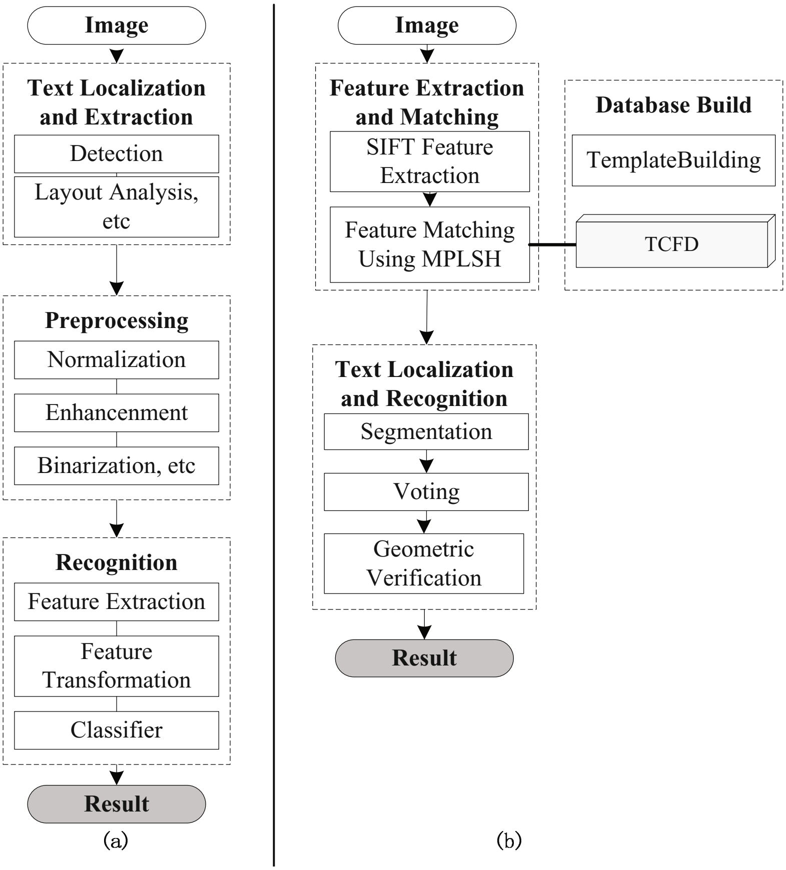
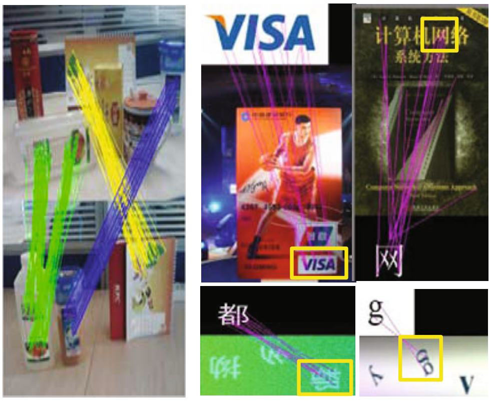


Fig. 1. Block diagram. (a) OCR-based framework; (b) Local feature-based framework. TCFD is template characters features database.



(a) (b)

Fig. 2. Local features matching. (a) Object matching; (b) Characters matching.

1.1 Related Work

There have been a number of successful text localization and recognition works reported in [6–11]. Most of them follow the OCR-based framework. Chen et al [7] reported an approach of detection and recognition of sign from natural scenes. Laplacian of Gaussian (LOG) edge detector, color modeling, layout analysis and affine rectification are used to detect text. Then normalization is used as preprocessing. At last, intensity-based OCR is applied to recognize the text. Koga et al [9] introduced a camera-based Kanji recognition system for mobile-phones. The first stage consists of 4 steps: preliminary binarization, coarse layout analysis, line direction detection and line segmentation. The second stage consists of another 4 steps to identify the text: fine binarization, pre-segmentation, character classification and post processing. More detailed surveys can be found in [10–12].

Our approach is most similar to the work of Campos [13], which utilizes local features and bag-of-visual-words model (BoW) to recognize single character in English and Kannada. Yet the main differences between these two approaches are quite clear: (1) Our approach could handle the detection and recognition of multiple characters other than single character; (2) Template images are machine-generated instead of manually collected in our approach, providing tremendous convenience for Chinese and Japanese text recognition.

2 Local Feature-Based Approach

Our framework (Figure 1(b)) consists of four stages: (1) Template images are obtained automatically via our template-build method, then template characters SIFT feature database (TCFD) is built. (2) The SIFT features of query image are extracted and matched to TCFD using MPLSH. (3) Multiple single-character-areas are located based on our segmentation algorithm. (4) For each single-character-area, a voting algorithm is used to identify candidate characters, which are then subjected to a geometric verification algorithm for final results. We describe these stages and methods in detail in the rest of this section.

2.1 Method of Building TCFD

Generation of template images for text matching is often challenged by the variation of characters (e.g. font, size, style). In some cases, the huge amounts of characters make the task even harder. For example, a total of 27474 characters are used in Chinese language compared with 26 letters in English.

In the field of image retrieval and object recognition, natural scene images are often used as template images. However, the local feature points in single character image are far less than that in a scene image. As a result, these interferences will greatly affect the matching accuracy if natural images are used as template images. Of note, for languages such as Chinese and Japanese, to obtain natural scene images will be indeed expensive and time consuming.

We applied the following strategies to build TCFD:

(1) The template images are machine-generated in monochrome mode without any additional noise and texture.
(2) According to fonts' similarity, a selected subset of fonts is used to generate template images per character.
(3) Every font per character will have two template images in TCFD (white-foreground/black-background or black-foreground/white-background) as shown in Figure 3(a). Using only one template image per font in some cases will result in zero matching points as shown in Figure 3(b). Furthermore, experimental results showed that using two template images readily gained 33.0% improvement over using one template image. Increasing the number of template images, however won't necessarily achieves further obvious improvement.

Table 1. Flowchart of Voting Algorithm

(1) Given an image, the initial vote of a candidate character is A, $A =$ the number of matched features in query image. $A = 3$ in Figure 4(a).

(2) Given a candidate character, B is the number of matched features in the candidate image. $B = 6$ Figure 4(a).

(3) A is not always equal to B due to the similar matching. Then, the final vote is $V = Min(A, B)$. $V = 3$ in Figure 4(a).

(4) The template characters with top C votes are identified as final candidate images, $C = R/V$, we have chosen to use $R = 60$ and C is limit from 2 to 20. The more V is identified, the fewer candidates are retrieved.

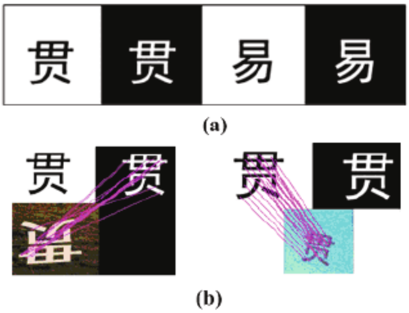


Fig. 3. Example of template images. (a) Two kinds of template images; (b) Only one template image has feature points matched for query images, so matching could be failed if only one template image is used.

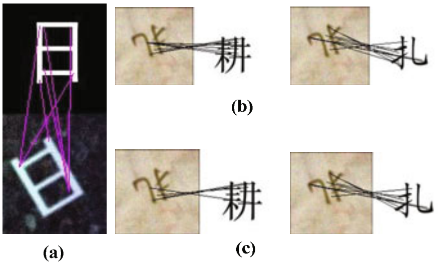


Fig. 4. Example images for voting algorithm and geometric verification algorithm. (a) Result before voting; (b) Result after voting; (c) Result after geometric verification.

2.2 Voting and Geometric Verification Algorithm

During text matching using local features, many mismatches can be caused by many factors, such as, similarities among characters, deformations and noises. Examples of mismatches could be found in Figure 4(a).

We designed a voting algorithm and gained 13.5% improvement. Optimized candidate characters are retrieved by using this algorithm. Flowchart of this algorithm is shown in Tab. 1.

Although the voting algorithm is helpful, there are still many mismatches in matching since local features are lack of global information. Such example results can be found in the left side of Figure 4(b).

Geometric verification can be used in character recognition for optimizing final results. This task however is often challenged by high computational cost and limited number of matched pairs.

Based upon the idea of pairwise constraint [16], we designed a geometric verification algorithm: Maximal Clique Matching for Text Recognition (MCM-TR). In MCM-TR, the global geometric constraint problem is expressed as the maximal clique problem in graph theory. MCM-TR starts from building a geometric correspondence graph (GCG) based upon the weak geometric constraint (WGC) information in local features. Then the global geometric relationship can be found by finding the maximal cliques in GCG. Given the characteristics of the global optimality of maximal cliques, MCM-TR is robust to occlusion, clutter, non-rigid deformations with the need of very few matched pairs.

We implemented MCM-TR as shown in Tab. 2 and achieved 9.5% improvement. The average matching time is 0.008 sec for two images with 60 matched pairs. Example results after geometric verification can be found in Figure 4(c).

Table 2. Flowchart of Geometric Verification Algorithm

(1) Given an image, and a candidate image identified by voting algorithm, all correspondences between these two images are labeled.

(2) For every matched pair, the space, scale and rotation information of SIFT features are extracted to estimate the WGC. To reduce the computational complexity, those match pairs whose points are not close in space and scale will be directly discarded.

(3) WGCs of all matched pairs are used to build GCG. GCG is an undirected and unweighted graph, in which each vertex represents a correspondence. The vertices are adjacent only when the correspondences are consistent with WGC. We make the projection from correspondences to GCG.

(4) The approximation algorithm proposed in [16] is extended to finding the maximal cliques in GCG. The maximal cliques just represent the global geometric relationship between the query image and the candidate image. To reduce the computational complexity, the maximal cliques containing too many or too few vertices are rejected.

(5) The candidate with max number of the matched pairs in the maximal clique is indentified as the final result.

2.3 Segmentation Algorithm and Multiple-Character Recognition

Segmentation algorithm is used to locate multiple single-character-areas in whole image. We call each area a sub-window, as Viola [17] use in face detection. We don't need to select all the feature points in the sub-window. The feature points of the same character are always similar in scale. We can filter the feature points by scale, which can greatly reduce the number of the local features. Furthermore, Hash table is used to rapidly obtain the local features in a sub-window.

Detail of algorithm is shown as following: Obtain the range of the location and scale of the local features matched in the MPLSH matching process. Let $W_{min} = S_{min}k$, $W_{max} = S_{max}k$. W_{min} and W_{max} represent the minimal and maximal size of the sub-window. The size of sub-windows increases by a factor of Δs between W_{min} and W_{max}. For each size, the sub-windows are shifted by some number of pixels $w\Delta l$. w is the size of sub-window. In each sub-window with size w, only those feature points whose scale is in the range of $(w/k, w\Delta s/k)$ are kept. The choice of Δl and Δd affects both the speed of recognition as well as accuracy. In this paper, $\Delta l = 2$ and $\Delta d = 0.5$.

Many sub-windows will be extracted in a query image. For example, in a 640x480 image, if the $W_{min} = 48$ and $W_{max} = 256$, 561 sub-windows will be extracted. It is high cost to recognize every sub-window.

The sub-windows will be subjected to the voting and geometric verification algorithm. However, it is not needed to recognize every sub-window. The reasons are: 1) there are few local features in some sub-windows. It is of low probability that there exist characters. 2) Some sub-windows cover the same region. If the characters in this region are recognized in one sub-window, the features of those characters do not need to be recognized anymore.

Table 3. Flowchart of Multiple Character Recognition Process

1. Statistic the number of the matched points in each sub-window. Those whose point number is less than threshold t will be removed from the heap. The sub-windows left are used to build a max heap.

2. While the point number of the top sub-window of the max heap is more than t:

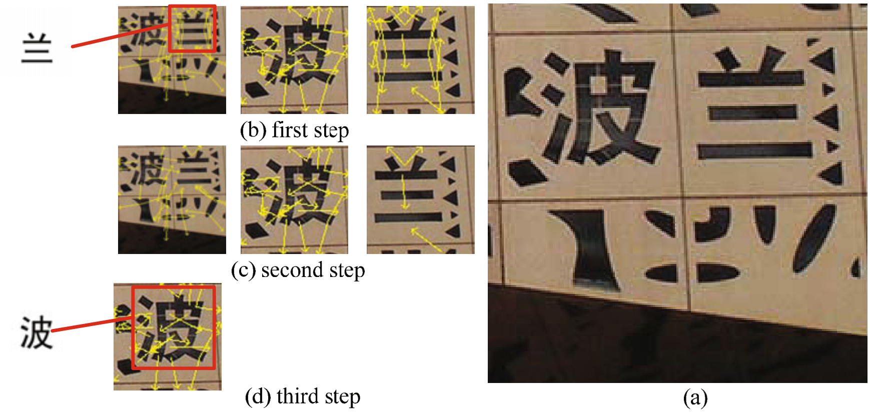
2.1 Recognize the top sub-window. Let define the recognized character is C and the point number is n.

2.2 If $n < t$ and n is less than half number of template character, we determine there is no character in the sub-window. The sub-window will be removed from the heap.

2.3 If C is recognized, the accurate region and orientation of C can be computed by the transformation between the character and the template. The points of C are removed from the query image. Then update the point number of the sub-windows that cover C.

2.4 Update the max heap.

3. Combine overlapped recognized characters. Only the character with the most feature points will be left.



(b) first step

(c) second step

(d) third step

(a)

Fig. 5. An example of multiple-character recognition. (a) is the query image containing two characters; Figures in (b), (c) and (d) are the sub-windows extracted from the query image. The yellow lines represent local features. The left sub-window is selected for recognition. In the first step, a character is successfully recognized. The local features of the character will be removed in every sub-window. In the second step, no character is recognized. The left sub-window will be removed. The right sub-window will be removed because of too few points. In the third step, the other character is correctly recognized. The selected sub-window will be removed because of too few points left.

The multiple characters recognition process is shown in Tab. 3. An example is shown in Figure 5.

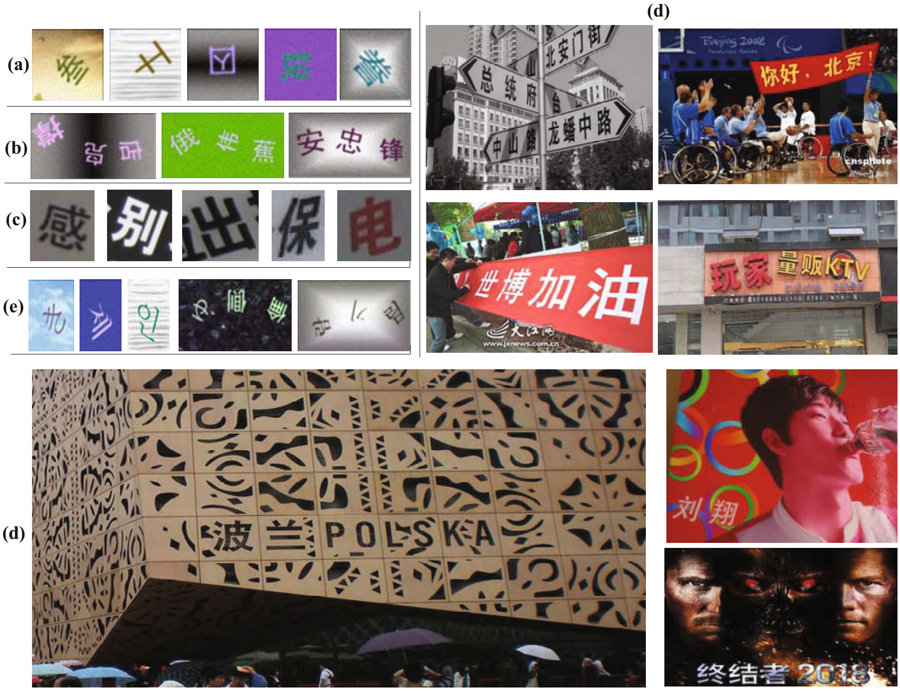


Fig. 6. Datasets. Examples of images used for the evaluation. (a) Dataset-A: Single Chinese characters; (b) Dataset-B: Multiple Chinese characters; (c) Dataset-C: Single Chinese characters from images of natural scenes; (d) Dataset-D: Multiple Chinese character images from natural scenes; (e) Dataset-E: Single and multiple characters of 3 languages scripts.

3 The Datasets

We built 5 datasets for the test: dataset-A, dataset-B, dataset-C and dataset-D are datasets containing only Chinese characters, dataset-E contains 3 language scripts (Chinese, Japanese and Korean). A summary of these five datasets is listed in Tab. 4. Examples of each dataset are shown in Figure 6.

4 Experimental Results

We performed 5 tests to evaluate our approach. Descriptions of these tests are shown in Tab. 4. In our experiments, we use Andrea Vedaldi's sift++[1] and Wei Dong's LSHKIT[2] for our SIFT and MPLSH [14] implementation.

The results of Test-1, Test-2, and Test-3 are shown in Tab. 5 and Tab. 7 accordingly. For Chinese text in complex scenes, we obtained average success

[1] http://www.vlfeat.org/~vedaldi/code/siftpp.html

[2] http://lshkit.sourceforge.net/index.html

Table 4. Description of each test

Tests	Datasets	Dataset Description	Objective of Test
Test-1	Dataset-A	3500 Single Chinese Characters; 168,000 machine-generated testing images in 12 fonts; 3 fonts in template characters.	To evaluate the success rate of single Chinese character, and describe the effect of various algorithms.
Test-2	Dataset-B	Multiple Chinese characters; 36,000 machine-generated test images in 12 fonts; 3 fonts in template characters.	To evaluate the success rate of multiple Chinese characters.
Test-3	Dataset-C	Single Chinese character image; Obtained from natural images; 1,000 testing images; 3 fonts in template characters.	To evaluate the success rate of single Chinese character from natural scenes, and compare to commercial OCR.
Test-4	Dataset-D	Multiple Chinese characters obtained from natural scenes; 120 Hei-like-font test images; 2 fonts in template characters.	To evaluate the success rate and false rate of multiple Chinese characters from natural scene images.
Test-5	Dataset-E	Chinese, Japanese, Korean; Single character and multiple characters; Machine generated; 15,700 testing images in 1 font; 1 font in template characters.	To evaluate the success rate of single character and multiple characters in 3 languages scripts.

rate of 77.3% (highest 94.1%) for single character and average success rate of 63.9% (highest 89.6%) for multiple characters. Compared with commercial OCR software, our approach improved the recognition accuracy by 12.9%. Given the character images all vary in fonts, lighting conditions, rotation, scale and affine deformation, these results are indeed encouraging.

There is 33.0% improvement by using our method to build the template images, 9.5% improvement by the geometric verification algorithm. The results demonstrated the efficiency of each steps of our approach.

It is also quite obvious from our studies that depending on the used fonts, the accuracy of text recognition changes dramatically too (e.g. the lowest rate 56.7% in the case of FangSong font). It indicated the importance of the selection of fonts in template images.

The results of Test-5 are shown in Tab. 6. Our approach also achieves encouraging results for language scripts other than Chinese script. We found the more SIFT points in a character images (the more complicated structure), the higher success rate.

The results of Test-4 are shown in Tab. 8. Results show that our approach is robustness in complex scenes. Some positive results are in Figure 7(a). We also found the success rate decreased for multiple characters from both natural scenes and machine-generated images. The main reasons are:

Table 5. Results of Test-1 and Test-2 (success rate of single Chinese character and multiple Chinese characters). E-1 is the approach without template-build method, E-2 is the approach without geometric verification algorithm. E-Single represents single character. E-Multi represents multiple characters.

Fonts	E-1	E-2	E-Single	E-Multi
Hei	53.3%	90.6%	94.1%	89.6%
MSYaHei	45.1%	70.4%	78.5%	63.2%
XiHei	49.4%	74.9%	84.3%	74.3%
PingHei	44.0%	65.4%	75.7%	66.7%
DengXian	48.8%	76.3%	84.6%	62.8%
YouYuan	37.1%	43.0%	58.7%	56.2%
GWArial	47.9%	80.6%	87.3%	66.1%
Song	44.5%	66.6%	76.4%	60.3%
FangSong	30.8%	40.8%	56.7%	50.7%
Kai	48.3%	81.2%	87.2%	67.8%
STKai	42.8%	67.0%	76.4%	60.2%
BWKai	39.7%	56.4%	67.5%	48.6%
Average	44.3%	67.8%	77.3%	63.9%

Table 6. Test results of Test-5

Language	Chinese	Japanese	Korean
Average number of SIFT points	60.3%	40.4%	28.5%
Success rate of single character	94.1%	88.3%	78.5%
Success rate of multiple characters	89.6%	77.1%	70.5%

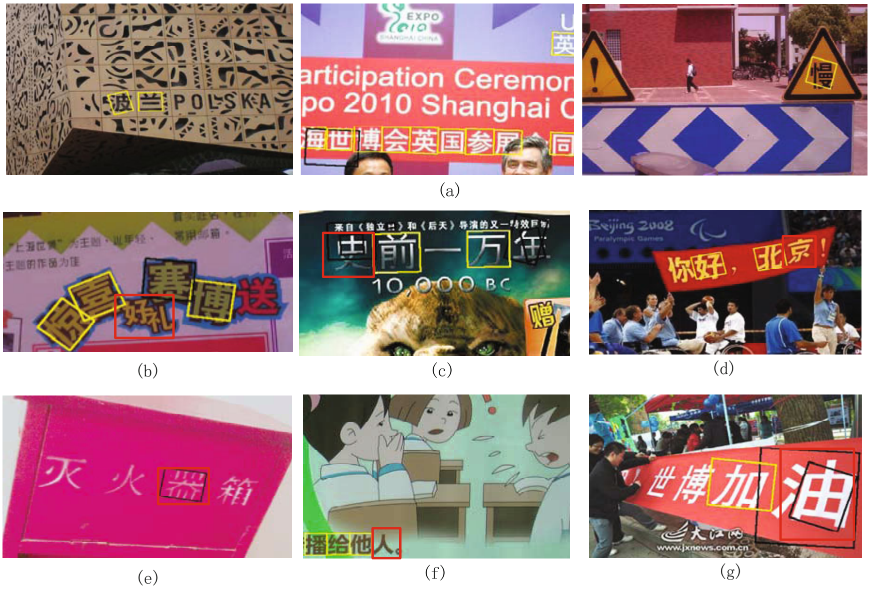
Table 7. Test results of Test-3

Methods	Our Approach	Hanwang-Wenhao OCR 7600	Tsinghua-OCR 9.0 Pro
Success Rate	73.1%	60.2%	44.7%

Table 8. Test results of Test-4. **SROC:** the rate of the correct recognized characters among all the characters. **FROC:** the rate of false characters among all the recognized ones. **SROI:** The rate of the images with all characters correctly recognized and no false ones. **FROI:** the rate of those images with no characters correctly recognized.

Language	SROC	FROC	SROI	FROI
Chinese	60.4%	30.6%	12.5%	15.8%

(1) The complex change in natural scenes, such as joined characters (Figure 7(b)), varied foreground (Figure 7(c)), shadows, vertical characters, outline characters, large deformation (Figure 7(d)) and large illumination change will lead



(a) (b) (c) (d) (e) (f) (g)

Fig. 7. Datasets. Examples of images used for the evaluation. (a) Dataset-A: Single Chinese characters; (b) Dataset-B: Multiple Chinese characters; (c) Dataset-C: Single Chinese characters from images of natural scenes; (d) Dataset-D: Multiple Chinese character images from natural scenes; (e) Dataset-E: Single and multiple characters of 3 languages scripts. Yellow rectangles represent correct recognition. Black ones represent false recognition. Red rectangles are manually drawn for further explanation.

to the great change in the scale, orientation and description of the feature points, which result in rejection of many matching pairs in both matching and geometric verification process. Moreover, low resolution (Figure 7(e)) and too thin strokes will cause very few SIFT feature detected.

(2) Threshold t will rejected characters with simple structures as well as the background noises. It is tradeoff between success rate and false rate. Figure 7(f) is the sample image that a simple character is rejected.

(3) Similar characters possibly received more votes than the character itself even after geometric verification, if wrong sub-windows are extracted and selected. In Figure 7(g), the selected sub-windows are either too big or too small.

5 Conclusion

In this paper, we describe a local-feature-based framework for text localization and recognition by only using SIFT features. The essential components in this framework include template-character-feature database buildup, a segmentation algorithm, a voting algorithm and a geometric verification algorithm. Our results demonstrated this approach performed well for texts in complex scenes, especially for those language scripts with complicated structures.

Although the robust performance of our approach suggests the local features matching can be utilized to address common problems in text localization and recognition, more works are needed toward a mature application. We plan to investigate other geometric verification methods and local features for better recognizing multiple characters with simple structure. We will explore such an approach in our future work.

Acknowledgement. This work was supported by National Natural Science Foundation of Shanghai (Grant No.10ZR1416400).

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: ICCV, vol. 2, pp. 91–110 (2004)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR, vol. 2, pp. 506–513 (2004)
4. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. In: CVPR, vol. 2, pp. 257–263 (2003)
5. Tuytelaars, T., Mikolajczyk, K.: A Survey on Local Invariant Features. In: Foundations and Trends in Computer Graphics and Vision (2008)
6. Chen, X., Yuille, A.: Detecting and Reading Text in Natural Scenes. In: CVPR, vol. 2, pp. 366–373 (2004)
7. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic detection and recognition of signs from natural scenes. IEEE Transactions on Image Processing 13, 87–99 (2004)
8. Chang, S.L., Chen, L.S., Chung, Y.C., Chen, S.W.: Automatic License Plate Recognition. IEEE Transactions on Intelligent Transportation Systems 5, 42–53 (2004)
9. Koga, M., Mine, R., Kameyama, T., Takahashi, T., Yamazaki, M., Yamaguchi, T.: Camera-based Kanji OCR for mobile-phones: practical issues. In: ICDAR (2005)
10. Liang, J., Doermann, D., Li, H.: Camera-based analysis of text and documents: A survey. IJDAR 7, 84–104 (2005)
11. Jung, K., Kim, K.I., Jain, A.K.: Text information extraction in images and video: a survey. Pattern Recognition 37, 977–997 (2004)
12. Fujisawa, H.: Forty years of research in character and document recognition - an industrial perspective. Pattern Recognition 41, 2435–2446 (2008)
13. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: VISAPP (2009)
14. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In: VLDB, pp. 950–961 (2007)
15. Johnson, D.S.: Approximation algorithms for combinational problems. JCSS 9, 256–278 (1974)
16. Leordeanu, M., Hebert, M.: A Spectral Technique for Correspondence Problems Using Pairwise Constraints. In: ICCV, vol. 2, pp. 1482–1489 (2005)
17. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: CVPR, pp. 511–218 (2001)