Recognition of Layout-Free Characters on Complex Background

Masakazu Iwamura, Takuya Kobayashi, Takahiro Matsuda and Koichi Kise
Graduate School of Engineering, Osaka Prefecture University
{masa, kise}@cs.osakafu-u.ac.jp, {kobayashi, matsuda}@m.cs.osakafu-u.ac.jp

***Abstract*—Recognizing characters in a scene is a challenging and unsolved problem. In this demonstration, we show an effective approach to cope with the problems: recognizing Japanese characters including complex characters such as Kanji (Chinese characters), which may not be aligned on a straight line and may be printed on a complex background. In the demo, our recognition method is applied to image sequences captured with a web camera. The recognition method is based on local features and their alignment. In addition, using a tracking method, recognition results and extracted features are accumulated so as to increase recognition accuracy as time goes on. The demo runs about 1 fps on a standard laptop computer.**

I. Introduction

It is a challenging and unsolved problem to let computers recognize characters. Characters may be complex (e.g., Japanese characters including Kanji (Chinese characters)), are not guaranteed to be aligned on a straight line, and may be printed on a complex background. Degradation of characters often happening in camera-captured images makes the problem more difficult. A recognition method is required to be reasonably fast. In this demonstration, we show an effective method to cope with these problems.

II. Method

This recognition method used in this demo is based on our past works [1], [2]. Due to limitation of space, a brief explanation about the method is given.

Figure 1 shows an overview of the recognition method. The recognition method is based on local features and their alignment, which are often used in the object recognition. The idea is that if the local features locate in the query image in the same alignment as ones in a reference image, the character of the reference image should exist in the region of the query image. The alignment of the features is confirmed by the local RANSAC algorithm, which is a variant of RANSAC algorithm [3] and in which the RANSAC algorithm is applied to a local region. As the local feature, we employ ASIFT [4] (a variant of SIFT [5] robust to Affine deformation).

III. Demo

The recognition process takes about 1 fps on a standard laptop computer. We cannot say 1 fps is fast enough with regard to usability. In order to improve response, the demo program consists of two threads, following the way used in [6]. One is dedicated to character recognition and the other is to tracking. Since the latter is much faster than the former, so as to realize quick response, the latter also handles capturing images and showing captured images with superimposing recognition results. There are two reasons to use tracking. One is to compensate for the change of the camera position and pose between a frame image used for recognition and one that recognition results are displayed. Before the recognition results are displayed, they are deformed using the homography between the two frames, which can be calculated using the tracking method. The other reason is to accumulate recognition results and extracted features for better results. This enables us to increase recall and precision as time goes on.

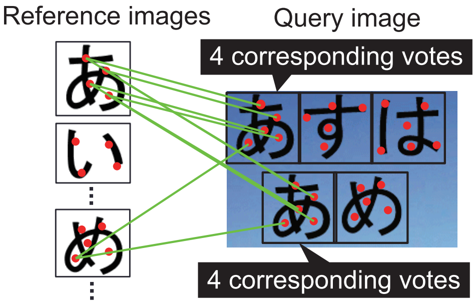


Fig. 1. An overview of the proposed method. Red points represent local features extracted and green lines do correspondences of features. Recognition results (characters and their bounding boxes) are determined at once based on correspondences of local features and their alignment. Many correspondences are omitted for better looking.

IV. Results

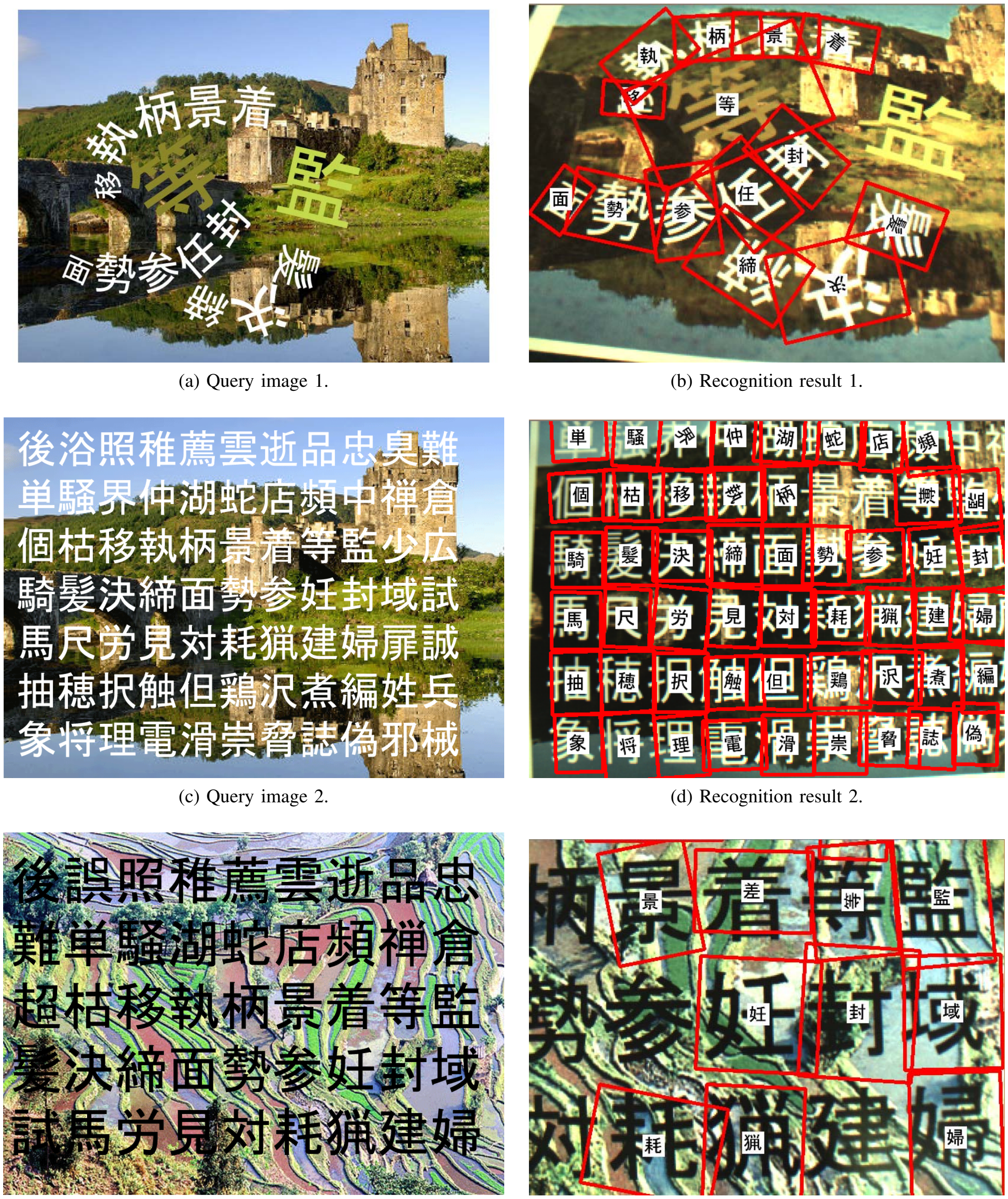
Some results are shown in Fig. 2. In each row, the original query image is shown on the left and the corresponding recognition result is shown on the right. The recognition results were obtained by capturing printed query images with a web camera and the recognition method is applied to the images. Red rectangles represent bounding boxes of recognized characters and recognition results were superimposed on them.

Acknowledgment

This work was supported in part by JST CREST project and JSPS KAKENHI Grant Number 25240028.

References

[1] M. Iwamura, T. Kobayashi, and K. Kise, “Recognition of multiple characters in a scene image using arrangement of local features,” *Proc. 11th Int’l Conf. on Document Analysis and Recognition (ICDAR2011)*, pp. 1409–1413, 2011.



(a) Query image 1.

(b) Recognition result 1.

(c) Query image 2.

(d) Recognition result 2.

(e) Query image 3.

(f) Recognition result 3.

Fig. 2. Some examples that were recognized by the demo system. Recognition results were obtained by capturing a part of query images.

[2] T. Kobayashi, M. Iwamura, and K. Kise, “An anytime algorithm for faster camera-based character recognition,” *Proc. 12th Int’l Conf. on Document Analysis and Recognition (ICDAR2013)*, 2013.

[3] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[4] J. Morel and G.Yu, “ASIFT: A new framework for fully affine invariant image comparison.” *SIAM Journal on Imaging Sciences*, vol. 2, 2009.

[5] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int’l Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[6] K. Takeda, K. Kise, and M. Iwamura, “Real-time document image retrieval on a smartphone,” in *Proc. 10th Int’l Workshop on Document Analysis Systems (DAS2012)*, 2012, pp. 225–229.