2012 10th IAPR International Workshop on Document Analysis Systems

Recognizing Words in Scenes with a Head-Mounted Eye-Tracker

Takuya Kobayashi*, Takumi Toyama†, Faisal Shafait†, Masakazu Iwamura*, Koichi Kise* and Andreas Dengel†
*Graduate School of Engineering Osaka Prefecture University, 1-1 Gakuencho, Naka, Sakai 599-8531 Japan
Email: kobayashi@m.cs.osakafu-u.ac.jp, {masa, kise}@cs.osakafu-u.ac.jp
†German Research Center for Artificial Intelligence (DFKI), Trippstadter Straße 122, 67663 Kaiserslautern, Germany
Email: <firstname.lastname>@dfki.de

***Abstract*—Recognition of scene text using a hand-held camera is emerging as a hot topic of research. In this paper, we investigate the use of a head-mounted eye-tracker for scene text recognition. An eye-tracker detects the position of the user's gaze. Using gaze information of the user, we can provide the user with more information about his region/object of interest in a ubiquitous manner. Therefore, we can realize a service such as the user gazes at a certain word and soon obtain the related information of the word by combining a word recognition system with eye-tracking technology. Such a service is useful since the user has to do nothing but gazes at interested words. With a view to realize the service, we experimentally evaluate the effectiveness of using the eye-tracker for word recognition. The initial results show the recognition accuracy was around 70% in our word recognition experiment and the average computational time was less than one second per a query image.**

***Keywords*-camera-based character recognition, scene images, local features, eye-tracking, gaze detection;**

I. Introduction

A camera-based character recognition system has many possibilities to help our daily life [1], [2], [3]. One good example is so-called translation camera system. The system recognizes a text in scenes and provides the user with translated words only by taking a picture of the words. Such a type of application is quite helpful especially when you are in a foreign country and surrounded by a huge number of unknown words. One of the existing methods which can be used in such a type of application was proposed by Iwamura et al. [4]. The method recognizes words in the query image with high accuracy in real-time. Besides, it provides information about the recognized words to the user with multiple forms such as translated meaning, an image related to the word, and so on. However, this system requires the user to hold the camera and direct the lens toward the words he/she is interested in. This constraint limits the usability of the application.

One solution is to use a head-mounted camera. A character recognition system was proposed that uses a head-mounted camera to capture images [5]. With this system, the user can obtain additional information of the interested word by directing the lens of the head-mounted camera to the word. Since this system does not require the user to hold a camera, it has less constraints than using a hand-held camera. However, this system has a problem that there is often a gap between the gaze point of the user and the direction of the user's head. Therefore, when the user likes to get the information about a certain word, he/she has to direct his/her head toward it. This might bother the user.

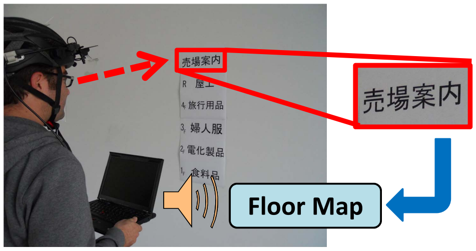


Figure 1. Translation camera system with a head-mounted eye-tracker.

In order to obtain the gaze position of the user, eye-tracking technology was developed. The user wears a head-mounted device that has two cameras. One captures the eye image and the other captures the scene image. This system provides the gaze position of the user in the captured image. Toyama et al. proposed an application called Museum Guide 2.0 to guide visitors in a museum by combining an object recognition technology with an eye-tracker [6]. When a visitor gazes at any exhibit, the application recognizes the exhibit and plays an audio file that provides additional information to the user about the exhibit. According to their experimental result, when we use the gaze information, the recognition accuracy is improved. Besides, there are two merits to use gaze information. First, it is useful to realize intuitive applications. Because people usually move their eyes instead of moving their head when they look at an interesting object. Second, we can also reduce the computational cost of the recognition system by using gaze information. Since we can obtain the gaze point, we can apply the recognition process to the neighbor region of the point.

In this paper, we evaluate the effectiveness of using an eye-tracking system for word recognition in scenes with a view to realize a translation camera system. Figure 1 shows a sample scenario to use the translation camera system in scenes. Since eye-tracking technology is still fresh, it has

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.74

never been used for a word recognition task in scenes. Thus, investigating how effectively the eye-tracker works on the task is important. In character recognition process, we used a method proposed by Iwamura et al. [7]. Their method recognizes characters by using SIFT [8]. We propose a word recognition method based on their character recognition method. In order to evaluate the word recognition method, we conduct two experiments. One is to optimize the parameters of the system. The other is to evaluate the recognition accuracy and the computational time of the method. In all the experiments, we used Japanese as a query language to realize a translation camera system from Japanese to other languages.

II. Method

In this section, we describe our word recognition method. The workflow is as follows: when the user gazes at words, the system crops the captured image so that the gaze point is the center of the cropped image. Then, local features are extracted from the image and the characters in the image are recognized by matching the local features. Recognized characters are then connected to their adjacent characters in order to obtain words. We describe more details of each process below.

A. Image Cropping by Using Eye-Tracking System

In order to obtain gaze information, we used SMI iViewX$^{\mathrm{TM}}$ HED as a head-mounted eye tracker in our experiments. This eye-tracker has two cameras. One is for capturing an eye image and the other is for capturing a scene image. The eye movement observed in the eye camera is analyzed by an eye-tracking algorithm provided by SMI to obtain the gaze position. Then, we crop the scene image so that the gaze point is the center of the image. By cropping the scene image, we can reduce the computational cost of the recognition process. However, preferably the size of a cropped image is large enough to contain all characters in a word. In addition, we magnify the cropped image by using bi-linear interpolation to enlarge the size of captured characters. Since characters in a scene image get smaller as the distance gets larger and when the size of characters is too small, the stability of local features decreases. Because characters must be distinguished by only their shape, extracting discriminative features from reduced-size character images is quite difficult.

B. Character Recognition Method

We extend the character recognition method proposed by Iwamura et al. [7]. Their method uses local features to recognize characters. These local features are extracted by using SIFT [8]. SIFT is invariant to changes of scale and rotation. In this paper, we adopted the affine-invariant version of SIFT (ASIFT) to extract features also robust to perspective transformation [9]. First, the proposed method extracts local features from local regions of a query image. Then each feature is matched to the most similar feature extracted from reference character images. In order to reduce the computational time, we use an approximate nearest neighbor search method proposed by Sato [10]. If only one character is in the query image, it can be recognized by using a simple voting method. A vote is cast for each reference character whose local feature is corresponding to a local feature from the query image. Then, the reference character which has the largest number of votes is returned as the recognition result. However, a query image usually has many characters. In order to recognize multiple characters at the same time, we use arrangements of local features extracted from each character to estimate the region of each character in the query image. Specifically, three pairs of matched feature points are used to calculate an affine matrix to project the character region upon the query image. Each character region is marked with a bounding box. The bounding boxes are projected according to the estimated affine transformation matrix. After all character regions are estimated, we can apply the simple voting method to each character region. A score for each character is given by

$$\mathrm{score} = \frac{m_p}{\sqrt{r_p}}, \tag{1}$$

where m_p is the number of feature points matched to the recognized character inside the character region and r_p is the number of feature points extracted from the reference image of the recognized character. r_p is used to normalize the difference between the number of feature points extracted from each reference character. Since the projected character region sometimes largely overlap with each other, we group such characters. Overlapped character regions are grouped if they satisfy the inequality given by

$$\mathrm{dist} < \mathrm{mean\_length}/2, \tag{2}$$

where dist is the distance between the center of two character regions and $\mathrm{mean\_length}$ is the average length of each side of the two bounding boxes. After the process, the recognized character with the highest score among them is treated as the recognition result in the group. Generally, the character recognition process finished in less than one second.

C. Word Recognition Method

Recognized characters in the query image are then connected with their adjacent characters to obtain words. Certain two characters are connected if they satisfy the inequality given by

$$\mathrm{dist} < \mathrm{mean\_length} \times 1.2, \tag{3}$$

where the meaning of each word is the same as before. When characters are connected horizontally, they are read from left-to-right, and when vertically connected it is top-to-bottom.

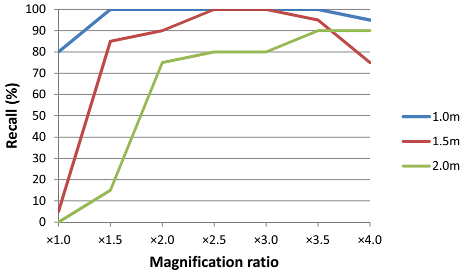


Figure 2. Relationship between magnification ratio and recall of character recognition.

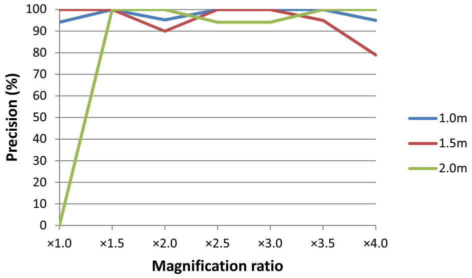


Figure 3. Relationship between magnification ratio and precision of character recognition.

III. Experiments

In this section, we show experimental results and discuss each of them. In order to evaluate the effectiveness to use eye-tracker for word recognition, we had two experiments. The first one is to optimize two parameters, the size of cropped image and the magnification ratio of the cropped image. The other one is to evaluate the recognition accuracy and computational time of the proposed word recognition system.

In the experiments, we employed 71 categories of Hiragana, 71 categories of Katakana and 1,945 categories of Kanji (Chinese character) in MS Gothic font for reference characters with the same condition as [7]. The resolution of our head-mounted camera of the eye-tracker was 752 × 480. All experiments were performed on a computer with Intel Core i5 2.53GHz CPU and 6GB memory.

A. Parameter Optimization

To optimize our word recognition system, we compared the performance of the system with changing two parameters related to the size of a query image. The first parameter is the size of an image cropped from a captured scene image and the other parameter is the magnification ratio of a cropped image. Since there is a trade-off relationship between these two parameters and the computational time, we need to select the best parameters.

Table I
Relationship between the distance from people to captured characters and the length on each side of a bounding square of a captured character.

Distance	1.0 m	1.5 m	2.0 m
Length (pixels)	45	30	25

Table II
Relationship among the size of a cropped image and magnification ratio and computational time (ms) to recognize characters in an image.

		Size of a Cropped Image (pixels)		
		200×200	250×250	300×300
Magnification Ratio	×2.5	813.2	855.0	998.2
	×3.0	874.8	1118.0	1424.3
	×3.5	1073.7	1473.8	1790.8

First, we select the well-balanced magnification ratio. Before we started the experiment, we investigated the relationship between the size of a character and typical distances from people to the characters to know how far people look at characters from. We asked five persons to look at words on a wall from the distance they feel natural to look at them. We prepared 6 words including 20 characters in total and the length of each side of the bounding box for each character was 6 centimeters. As a result, the range of the distance was approximately between 1 and 2 meters. Thus, we investigated the accuracy of character recognition when the characters were captured from 1.0, 1.5 and 2.0 meters distance, respectively.

Figures 2 and 3 show the relationships between the magnification ratio and the recall / precision of character recognition for each distance. Recall and precision are calculated by

$$\text{recall} = \frac{c_r}{n_c}, \quad \text{precision} = \frac{c_r}{n_r}, \tag{4}$$

where c_r is the number of correctly recognized characters, n_c is the number of characters on the wall and n_r is the number of recognized characters including correct and incorrect recognition. For each distance, recognition accuracy increased as the images were digitally magnified. However, the recognition accuracy of 1.0 and 1.5 meters decreases when the magnification ratio reaches 4.0. This is because when an image is magnified too large, the image is blurred and the stability of local features declines. Table I shows the relationship between the distances from people to the characters and the length on each side of a bounding box for a captured character. We rounded the numbers to the nearest multiples of 5. By investigating how the length of each side of a character and the magnification ratio affected the recognition accuracy, we found out that the length should be more than 60 pixels to achieve over 80% recall rates. For example, when the distance was 2.0 meters, we need to magnify the image 2.5 times to exceed the

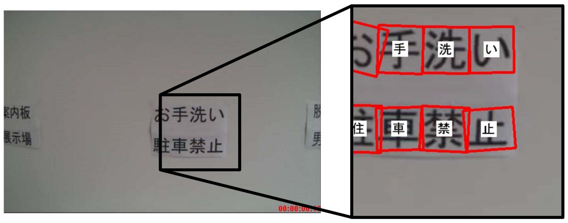
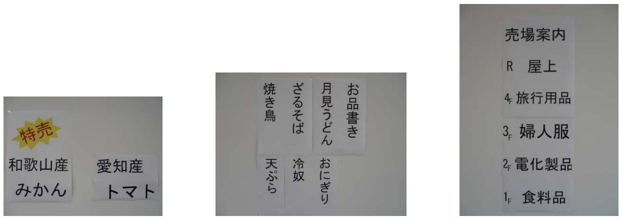


Figure 4. Failure case of recognition result. Because the size of the cropped image was small, some characters were not contained in the image completely.

Table III
OVERALL RECOGNITION ACCURACY OF THE CHARACTER AND WORD RECOGNITION SYSTEM WITH A HEAD-MOUNTED EYE-TRACKER

Angle	0°	30°
Recall (%)	88.1	69.3
Precision (%)	94.3	90.4
Recall(word) (%)	69.7	42.8



(a) Supermarket (b) Menu in Restaurant (c) Floor Map of Department Store

Figure 5. Three kinds of situations in Japan. We prepared them to simulate real cases.

Table IV
COMPARISON OF THE COMPUTATIONAL TIME BETWEEN WITH AND WITHOUT GAZE INFORMATION. WHEN WE USE GAZE INFORMATION, WE CAN CROP THE CAPTURED IMAGE AND REDUCE THE COMPUTATIONAL TIME.

	With Gaze Information	Without
Computational Time (ms)	917.0	3101.2

length of 60 pixels. From these results, we decide to select a magnification ratio from 2.5, 3.0, and 3.5.

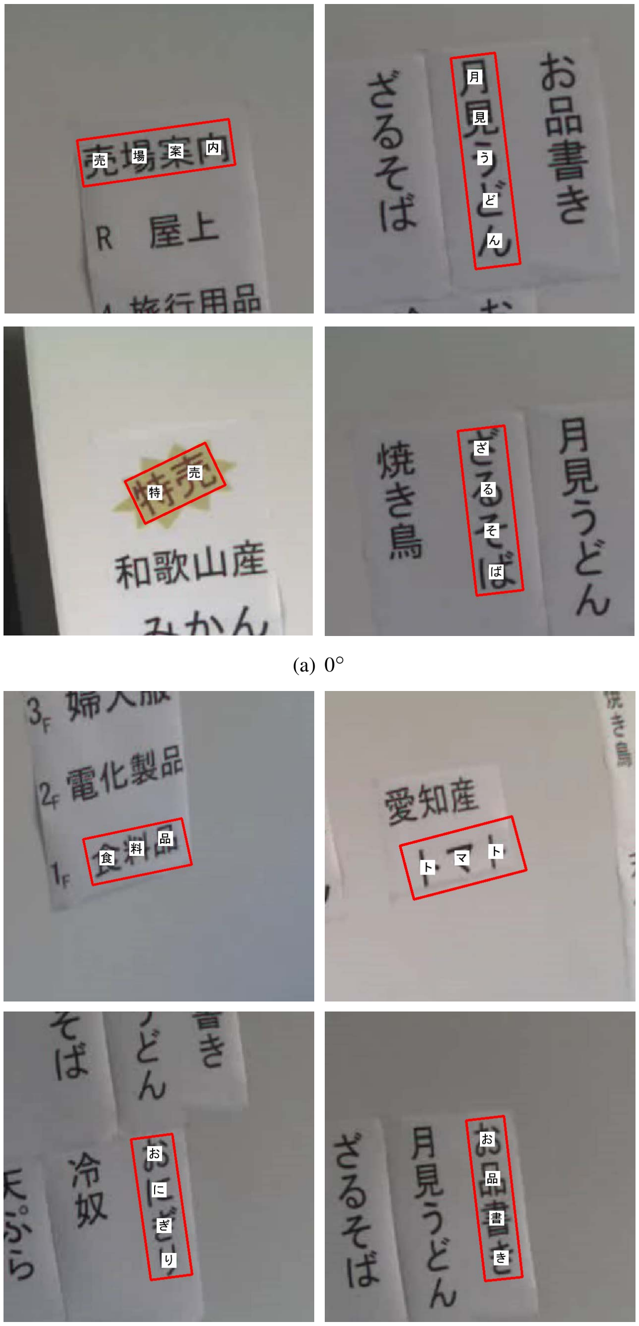
In order to find the best combination of the magnification ratio and the size of a cropped image, we conducted another experiment. We then investigated how the size of cropped image and magnification ratio affect the computational time. Table II shows the relationship between them. Computational time shown in the table was measured as the time needed to recognize characters in an image.

From this result and Figs. 2 and 3, one might think 200 × 200 pixels seem better with respect to the computational time. However, the size was sometimes too small to contain all characters in a word when images were captured from 1.0 meter distance as shown in Fig. 4. Therefore, we selected the combination of parameters that the size of a cropped image was 250 × 250 pixels and the magnification ratio was 2.5 since the computational time did not reach one second. We used these parameters in the following experiment.

B. Evaluation of the Word Recognition System

Next, we conducted another experiment to evaluate our word recognition system. We asked 13 persons to look at words on a wall as they usually do so. We set the distance between the wall and the persons as 1.5 meters and they looked at the words from two viewpoints, straight in front of the wall 0° and 30° left from that point. Figure 5 shows three kinds of situations we prepared to simulate the real scenes in Japan. They contained 18 Japanese words and 60 characters in total and the length of each side of a bounding box for a character ranged from 5.5 to 7.0 centimeters. First, we calibrated the eye-tracker by asking the user to look at five points on the wall. Then, we asked each of them to gaze at each word for several seconds. We recorded the video files for every word and then applied the character and word recognition process only to the fixated frames by the eye-tracker. Ten frames were used for each word and we calculated the average of the recognition results. In the experiment, we treated only one recognized word which was closest to the gaze point as the recognition result. Table III shows the recall and precision of character and word recognition calculated from whole recognition results. We achieved a high recall rate for character recognition with the angle of 0°. Although the recall decreased when the angle was 30°, the precision for both angles was over 90%. The drop of the recall was caused by changing a parameter of ASIFT descriptor. We can choose which to prefer, a robustness to perspective distortion or a reduction of computational time by changing the parameter. Since we selected the latter in the experiment, the recall decreased when the angle was 30°. An analysis of the recorded gaze data showed that almost all gaze positions were on the correct query word. Only when a user gazed at words which were much lower than their eyes, the gaze positions sometimes pointed to the wrong word. As shown in Table IV, when we use the gaze information, the computational time was three times as fast as the time without it. When we did not use the gaze information, we used the entire image without cropping since there was no information which region to crop. Regarding the size of a cropped image, only when the user gazed at the edge of a long word, the system failed to contain the whole word region into a cropped image. This problem can be solved by accumulating the information of recognized characters through several frames as we discuss later in this section. From these results, we confirmed we can improve the performance by using the gaze information.

Next, we consider the word recognition accuracy. Figure 6



(a) 0°

(b) 30°

Figure 6. Examples of correct word recognition results.

shows examples of correct word recognition result. A red bounding box is the region of the word and recognized characters are put on the center of each character region. For both angles, the recognition accuracy decreased compared with the results of character recognition. There are two reasons of this result.

First, in order to detect a word region, we connected adjacent characters. Thus, when the method fails to detect a character in the middle of a word, it cannot connect the separated parts of a word. To solve this problem, it would be effective to use a word segmentation approach. MSER can be used to detect word region as used in [5], [11], [12]. By combining MSER with gaze information, we might reduce the computational time. Besides, in order to improve the recognition accuracy we consider to accumulate feature points and recognition results through several frames when we realize a translation camera system. If the user gazes at any interesting word for several seconds, the system can accumulate the recognition results through the several frames. This method can recognize long words even if they are not contained in a cropped image completely. By using the KLT feature tracker [13] to track character regions in a captured image, we can realize such a process.

The second problem was that we recognized only one character per a character region in our method. As explained in Section 2, if regions of recognized characters overlap with each other, we treated the character which has the highest score among them as the recognition result. However, since many Japanese characters have similar shape with each other, such characters were often recognized as their similar characters. Thus, considering the rest of detected characters in a character region is necessary. A simple way is to create a candidate character lattice from the detected characters in a word. We can find the best combination of characters to be a proper word by considering the scores or by comparing with the list of words in a dictionary.

Finally, we discuss the usability of an eye-tracker. Through the experiments, we confirmed that the user can indicate the gaze point exactly. However, there is a constraint that the gaze point must be contained in a captured image. Thus, the object the user is gazing at might not be contained in a captured image when the angle between the line of sight and the direction of the scene camera is too large. Though such a case might rarely occur, we need a research about the frequency of it. One solution of the problem is to use a pan-tilt-zoom camera as the scene camera. We can mount the camera on an eye-tracker and make the camera follow the eye movement of the user.

IV. Conclusion

In this paper, we evaluated the effectiveness of using an eye-tracking system for word recognition in scenes. By using an eye-tracker, we can point to our interested word by gazing at it. Since gaze action is quite natural for humans, this system can improve the usability of applications. With a view to realize a translation camera system with a head-mounted eye-tracker, we had experiments to confirm the performance when we use the gaze information for the word recognition. As a result, we had 69.7% recall in our word recognition experiment. The gaze information worked well to point a certain word correctly and we could reduce the computational time by using the gaze information to crop the captured scene image. In order to realize a translation camera application, we need to improve the recognition accuracy, as well as reducing the computational time of character recognition process. Our future work is to realize the translation camera system with our word recognition method and to evaluate the usability of it by comparing with other methods.

ACKNOWLEDGMENT

This work was supported in part by CREST project from JST, Kayamori Foundation of Informational Science Advancement, and the Grant-in-Aid for Young Scientists (B) (21700202) from Japan Society for the Promotion of Science (JSPS).

REFERENCES

[1] M. Iwamura, T. Tsuji, and K. Kise, "Real-life clickable text," *SPIE Newsroom*, Dec. 2010. [Online]. Available: http://spie.org/x43601.xml

[2] X. Shi and Y. Xu, "A wearable translation robot," in *Proc. of Int. Conf. on Robotics and Automation*, 2005, pp. 4400–4405.

[3] Y. Watanabe, Y. Okada, Y.-B. Kim, and T. Takeda, "Translation camera," in *Proc. of Int. Conf. on Pattern Recognition*, vol. 1, Aug. 1998, pp. 613–617.

[4] M. Iwamura, T. Tsuji, and K. Kise, "Memory-based recognition of camera-captured characters," in *Proc. of Int. Workshop on Document Analysis Systems*, Jun. 2010, pp. 89–96.

[5] C. Merino-Gracia, K. Lenc, and M. Mirmehdi, "A head-mounted device for recognizing text in natural scenes," in *Proc. of Int. Workshop on Camera-based Document Analysis and Recognition*, Sep. 2011, pp. 27–32.

[6] T. Toyama, T. Kieninger, F. Shafait, and A. Dengel, "Museum guide 2.0 - an eye-tracking based personal assistant for museums and exhibits," in *Proc. of Int. Conf. on Re-Thinking Technology in Museums*, May 2011.

[7] M. Iwamura, T. Kobayashi, and K. Kise, "Recognition of multiple characters in a scene image using arrangement of local features," in *Proc. of Int. Conf. on Document Analysis and Recognition*, Sep. 2011, pp. 1409–1413.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Jour. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[9] J. Morel and G.Yu, "Asift: A new framework for fully affine invariant image comparison." *SIAM Jour. on Imaging Sciences*, vol. 2, Apr. 2009.

[10] T. Sato, M. Iwamura, and K. Kise, "Fast approximate nearest neighbor search based on improved approximate distance," in *Proc. of the Institute of Electronics, Information and Communication Engineers*, vol. 111, no. 193, Sep. 2011, pp. 61–66.

[11] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. of the British Machine Vision Conference*, vol. 1, 2002, pp. 384–393.

[12] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," *Proc. 10th Asian Conference on Computer Vision*, pp. 770–783, 2011.

[13] J. Shi and C. Tomasi, "Good features to track," in *IEEE Proc. of Computer Vision and Pattern Recognition*, 1994, pp. 593–600.