Portable Translator Capable of Recognizing Characters on Signboard and Menu Captured by Built-in Camera

Hideharu Nakajima, Yoshihiro Matsuo, Masaaki Nagata, Kuniko Saito
NTT Cyber Space Laboratories, NTT Corporation
Yokosuka, 239-0847, Japan
{nakajima.hideharu, matsuo.yoshihiro, nagata.masaaki, saito.kuniko}
@lab.ntt.co.jp

Abstract

We present a portable translator that recognizes and translates phrases on signboards and menus as captured by a built-in camera. This system can be used on PDAs or mobile phones and resolves the difficulty of inputting some character sets such as Japanese and Chinese if the user doesn't know their readings. Through the high speed mobile network, small images of signboards can be quickly sent to the recognition and translation server. Since the server runs state of the art recognition and translation technology and huge dictionaries, the proposed system offers more accurate character recognition and machine translation.

1 Introduction

Our world contains many signboards whose phrases provide useful information. These include destinations and notices in transportation facilities, names of buildings and shops, explanations at sightseeing spots, and the names and prices of dishes in restaurants. They are often written in just the mother tongue of the host country and are not always accompanied by pictures. Therefore, tourists must be provided with translations.

Electronic dictionaries might be helpful in translating words written in European characters, because key-input is easy. However, some character sets such as Japanese and Chinese are hard to input if the user doesn't know the readings such as *kana* and *pinyin*. This is a significant barrier to any translation service. Therefore, it is essential to replace keyword entry with some other input approach that supports the user when character readings are not known.

One solution is the use of optical character recognition (OCR) (Watanabe et al., 1998; Haritaoglu, 2001; Yang et al., 2002). The basic idea is the connection of OCR and machine translation (MT) (Watanabe et al., 1998) and implementation with personal data assistant (PDA) has been proposed (Haritaoglu, 2001; Yang et al., 2002). These are based on the document OCR which first tries to extract character regions; performance is weak due to the variation in lighting conditions. Although the system we propose also uses OCR, it is characterized by the use of a more robust OCR technology that doesn't first extract character regions, by language processing to offset the OCR shortcomings, and by the use of the client-server architecture and the high speed mobile network (the third generation (3G) network).

2 System design

Figure 1 overviews the system architecture. After the user takes a picture by the built-in camera of a PDA, the picture is sent to a controller in a remote server. At the server side, the picture is sent to the OCR module which usually outputs many character candidates. Next, the word recognizer identifies word sequences in the candidates up to the number specified by the user. Recognized words are sent to the language translator.

The PDA is linked to the server via wireless com-

Proceedings of the ACL Interactive Poster and Demonstration Sessions,
pages 61–64, Ann Arbor, June 2005.

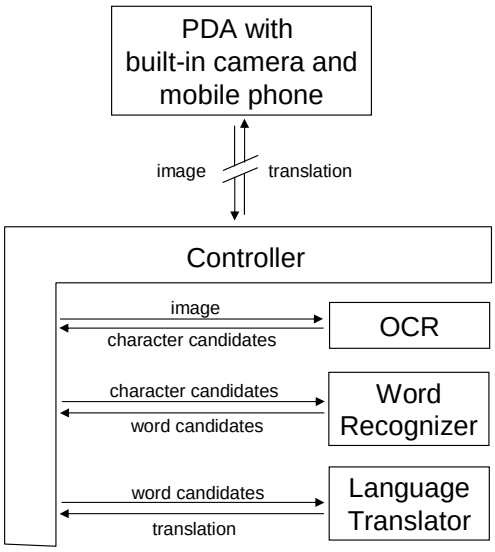

Figure 1: System architecture: http protocol is used between PDAs and the controller.

munication. The current OCR software is Windows-based while the other components are Linux programs. The PDA uses Windows.

We also implemented the system for mobile phones using the i-mode and FOMA devices provided by NTT-DoCoMo.

3 Each component

3.1 Appearance-based full search OCR

Research into the recognition of characters in natural scenes has only just begun (Watanabe et al., 1998; Haritaoglu, 2001; Yang et al., 2002; Wu et al., 2004). Many conventional approaches first extract character regions and then classify them into each character category. However, these approaches often fail at the extraction stage, because many pictures are taken under less than desirable conditions such as poor lighting, shading, strain, and distortion in the natural scene. Unless the recognition target is limited to some specific signboard (Wu et al., 2004), it is hard for the conventional OCR techniques to obtain sufficient accuracy to cover a broad range of recognition targets.

To solve this difficulty, Kusachi et al. proposed a robust character classifier (Kusachi et al., 2004). The classifier uses appearance-based character reference pattern for robust matching even under poor capture conditions, and searches the most probable

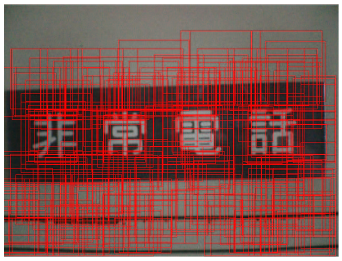

Figure 2: Many character candidates raised by appearance-based full search OCR: Rectangles denote regions of candidates. The picure shows that candidates are identified in background regions too.

region to identify candidates. As full details are given in their paper (Kusachi et al., 2004), we focus here on just its characteristic performance.

As this classifier identifies character candidates from anywhere in the picture, the precision rate is quite low, i.e. it lists a lot of wrong candidates. **Figure 2** shows a typical result of this OCR. Rectangles indicate erroneous candidates, even in background regions. On the other hand , as it identifies multiple candidates from the same location, it achieves high recall rates at each character position (over 80%) (Kusachi et al., 2004). Hence, if character positions are known, we can expect that true characters will be ranked above wrong ones, and greater word recognition accuracies would be achieved by connecting highly ranked characters in each character position. This means that location estimation becomes important.

3.2 Word recognition

Modern PDAs are equipped with styluses. The direct approach to obtaining character location is for the user to indicate them using the stylus. However, pointing at all the locations is tiresome, so automatic estimation is needed. Completely automatic recognition leads to extraction errors so we take the middle approach: the user specifies the beginning and ending of the character string to be recognized and translated. In **Figure 3**, circles on both ends of the string denote the user specified points. All the locations of characters along the target string are estimated from these two locations as shown in Figure 3 and all the candidates as shown in Figure 2.

Figure 3: Two circles at the ends of the string are specified by the user with stylus. All the character locations (four locations) are automatically estimated.

3.2.1 Character locations

Once the user has input the end points, assumed to lie close to the centers of the end characters, the automatic location module determines the size and position of the characters in the string. Since the characters have their own regions delineated by rectangles and have x,y coordinates (as shown in Figure 2), the module considers all candidates and rates the arrangement of rectangles according to the differences in size and separation along the sequences of rectangles between both ends of the string. The sequences can be identified by any of the search algorithms used in Natural Language Processing like the forward Dynamic Programming and backward A* search (adopted in this work). The sequence with the highest score, least total difference, is selected as the true rectangle (candidate) sequence. The centers of the rectangles are taken as the locations of the characters in the string.

3.2.2 Word search

The character locations output by the automatic location module are not taken as specifying the correct characters, because multiple character candidates are possible at the same location. Therefore, we identify the words in the string by the probabilities of character combinations. To increase the accuracy, we consider all candidates around each estimated location and create a character matrix, an example of which is shown in **Figure 4**. At each location, we rank the candidates according to their OCR scores, the highest scores occupy the top row. Next, we apply an algorithm that consists of similar character matching, similar word retrieval, and word sequence search using language model scores

1	2	3	4
非	吊	**電**	詰
乍	**常**	雷	**話**
匪	亮	雪	聞
罪	席	亀	諸
主	呈	巴	間
:	:	:	:

Figure 4: A character matrix: Character candidates are bound to each estimated location to make the matrix. Bold characters are true.

(Nagata, 1998).

The algorithm is applied from the start to the end of the string and examines all possible combinations of the characters in the matrix. At each location, the algorithm finds all words, listed in a word dictionary, that are possible given the location; that is, the first location restricts the word candidates to those that start with this character. Moreover, to counter the case in which the true character is not present in the matrix, the algorithm identifies those words in the dictionary that contain characters similar to the characters in the matrix and outputs those words as word candidates. The connectivity of neighboring words is represented by the probability defined by the language model. Finally, forward Dynamic Programming and backward A* search are used to find the word sequence with highest probability. The string in the Figure 3 is recognized as “非常電話.”

3.3 Language translation

Our system currently uses the ALT-J/E translation system which is a rule-based system and employs the multi-level translation method based on constructive process theory (Ikehara et al., 1991). The string in Figure 3 is translated into “Emergency telephones.”

As target language pairs will increased in future, the translation component will be replaced by statistical or corpus based translators since they offer quicker development. By using this client-server architecture on the network, we can place many task specific translation modules on server machines and flexibly select them task by task.

Table 1: Character Recognition Accuracies

[%]	OCR	OCR+manual	OCR+auto
recall	91	91	91
precision	12	82	80

4 Preliminary evaluation of character recognition

Because this camera base system is primarily for inputting character sets, we collected 19 pictures of signboards with a 1.2 mega pixel CCD camera for a preliminary evaluation of word recognition performance. Both ends of a string in each picture were specified on a desk-top personal computer for quick performance analysis such as tallying up the accuracy. Average string length was five characters. The language model for word recognition was basically a word bigram and trained using news paper articles.

The base OCR system returned over one hundred candidates for every picture. Though the average character recall rate was high, over 90%, wrong candidates were also numerous and the average character precision was about 12%.

The same pictures were evaluated using our method. It improved the precision to around 80% (from 12%). This almost equals the precision of about 82% obtained when the locations of all characters were manually indicated (**Table1**). Also the accuracy of character location estimation was around 95%. 11 of 19 strings (phrases) were correctly recognized.

The successfully recognized strings consisted of characters whose sizes were almost the same and they were evenly spaced. Recognition was successful even if character spacing almost equaled character size. If a flash is used to capture the image, the flash can sometimes be seen in the image which can lead to insertion error; it is recognized as a punctuation mark. However, this error is not significant since the picture taking skill of the user will improve with practice.

5 Conclusion and future work

Our system recognizes characters on signboards and translates them into other languages. Robust character recognition is achieved by combining high-recall and low-precision OCR and language processing.

In future, we are going to study translation qualities, prepare error-handling mechanisms for brittle OCR, MT and its combination, and explore new application areas of language computation.

Acknowledgement

The authors wish to thank Hisashi Ohara and Akihiro Imamura for their encouragement and Yoshinori Kusachi, Shingo Ando, Akira Suzuki, and Ken'ichi Arakawa for providing us with the use of the OCR program.

References

Ismail Haritaoglu. 2001. InfoScope: Link from Real World to Digital Information Space. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, Springer-Verlag, pages 247-255.

Satoru Ikehara, Satoshi Shirai, Akio Yokoo and Hiromi Nakaiwa. 1991. Toward an MT System without Pre-Editing - Effects of New Methods in ALT-J/E -. In *Proceedings of the 3rd MT Summit*, pages 101-106.

Yoshinori Kusachi, Akira Suzuki, Naoki Ito, Ken'ichi Arakawa. 2004. Kanji Recognition in Scene Images without Detection of Text Fields - Robust Against Variation of Viewpoint, Contrast, and Background Texture. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 204-207.

Masaaki Nagata. 1998. Japanese OCR Error Correction using Character Shape Similarity and Statistical Language Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 922-928.

Yasuhiko Watanabe, Yoshihiro Okada, Yeun-Bae Kim, Tetsuya Takeda. 1998. Translation Camera. In *Proceedings of the 14th International Conference on Pattern Recognition*, pages 613–617.

Wen Wu, Xilin Chen, Jie Yang. 2004. Incremental Detection of Text on Road Signs from Video with Application to a Driving Assistant System. In *Proceedings of the ACM Multimedia 2004*, pages 852-859.

Jie Yang, Xilin Chen, Jing Zhang, Ying Zhang, Alex Waibel. 2002. Automatic Detection and Translation of Text From Natural Scenes. In *Proceedings of ICASSP*, pages 2101-2104.