KooSHO: Japanese Text Input Environment based on Aerial Hand Writing

Masato Hagiwara
Rakuten Institute of Technology
215 Park Avenue South,
New York, NY, USA 10003
masato.hagiwara@mail.rakuten.com

Soh Masuko
Rakuten Institute of Technology
4-13-9 Higashi-shinagawa
Shinagawa-ku, Tokyo, JAPAN 140-0002
so.masuko@mail.rakuten.com

Abstract

Hand gesture-based input systems have been in active research, yet most of them focus only on single character recognition. We propose KooSHO: an environment for Japanese input based on aerial hand gestures. The system provides an integrated experience of character input, Kana-Kanji conversion, and search result visualization. To achieve faster input, users only have to input consonant, which is then converted directly to Kanji sequences by *direct consonant decoding*. The system also shows suggestions to complete the user input. The comparison with voice recognition and a screen keyboard showed that KooSHO can be a more practical solution compared to the existing system.

1 Introduction

In mobile computing, intuitive and natural text input is crucial for successful user experience, and there have been many methods and systems proposed as the alternatives to traditional keyboard-and-mouse input devices. One of the most widely used input technologies is voice recognition such as Apple Inc.'s *Siri*. However, it has some drawbacks such as being vulnerable to ambient noise and privacy issues when being overheared. Virtual keyboards[1] require extensive practice and could be error-prone compared to physical keyboards.

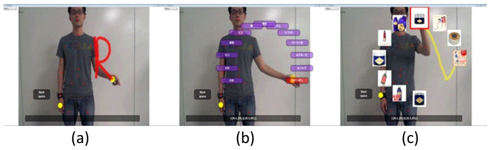

Figure 1: Overall text input procedure using KooSHO — (a) Character recognition (b) Kana-Kanji conversion results (c) Search results

In order to address these issues, many gesture-based text input interfaces have been proposed, including a magnet-based hand writing device (Han et al., 2007). Because these systems require users to wear or hold special devices, hand gesture recognition systems based on video cameras are proposed, such as Yoon et al. (1999) and Sonoda and Muraoka (2003). However, a large portion of the literature only focuses on single character input. One must consider overall text input experience when users are typing words and phrases. This problem is pronounced for languages which require explicit conversion from Romanized forms to ideographic writing systems such as Japanese.

In this paper, we propose *KooSHO*: an integrated environment for Japanese text input based on aerial hand gestures. It provides an integrated experience of character input, Kana-Kanji conversion, i.e., conversion from Romanized forms to ideographic (Kanji) ones, and search result visualization. Figure 1 shows the overall procedure using KooSHO. First, (a) a user draws alphabetical shapes in the air, whose hand position is captured by Microsoft Kinect. KooSHO then recognizes characters, and after

[1] http://www.youtube.com/watch?v=h9htRy0-sUw

Proceedings of the NAACL HLT 2013 Demonstration Session, pages 24–27,
Atlanta, Georgia, 10-12 June 2013. ©2013 Association for Computational Linguistics

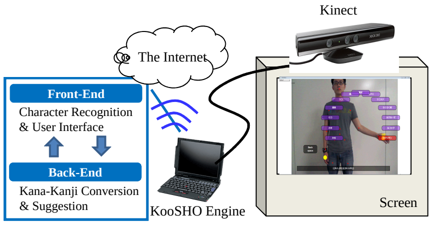
Kinect
The Internet
Front-End
Character Recognition & User Interface
Back-End
Kana-Kanji Conversion & Suggestion
KooSHO Engine
Screen

Figure 2: Configuration of the KooSHO system

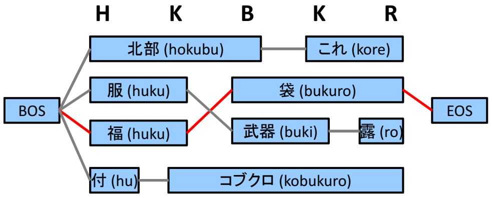

Figure 3: Lattice search based on consonants

Kana-Kanji conversion, the results are shown in a circle centered at the user's shoulder (b). Finally, the user can choose one of the candidates by "touching" it, and (c) the search result using the chosen word as the query is shown in circle again for the user to choose.

KooSHO has several novelties to achieve seamless yet robust text input, including:

Non-restrictive character forms — the system does not restrict on the input character forms, unlike Graffiti 2[2].

Robust continuous recognition and conversion — Aerial handwriting poses special difficulties since the system cannot obtain individual strokes. We solve this problem by employing a discriminative Kana-Kanji conversion model trained on the specific domain.

Faster input by suggestions and consonant input — KooSHO shows suggestions to predict the words the user is about to input, while it allows users to type only consonants, similar to Tanaka-Ishii et al. (2001). We propose *direct consonant decoding*, which runs Viterbi search directly on the input consonant sequence without converting them back into Kana candidates.

We conducted the evaluations on character recognition and Kana-Kanji conversion accuracy to measure KooSHO's performance. We also ran an overall user experience test, comparing its performance with the voice recognition software *Siri* and a screen keyboard.

2 Character Recognition

Figure 2 describes the overall configuration. A user draws alphabetical shapes in the air, which is captured by Kinect and sent to KooSHO. We used the skeleton recognition functionalities included in Kinect for Windows SDK v1.5.1. The system consists of the front-end and back-end parts, which are responsible for character recognition and user interface, and Kana-Kanji conversion and suggestion, respectively.

We continuously match the drawn trajectory to templates (training examples) using dynamic programming. The trajectory and the templates are both represented by 8 direction features to facilitate the match, and the distance is calculated based on how apart the directions are. This coding system is robust to scaling of characters and a slight variation of writing speed, while not robust to stroke order. This is repeated every frame to produce the distance between the trajectory ending at the current frame and each template. If the distance is below a certain threshold, the character is considered to be the one the user has just drawn.

If more than one characters are detected and their periods overlap, they are both sent as alternative. The result is represented as a lattice, with alternation and concatenation. To each letter a confidence score (the inverse of the minimum distance from the template) is assigned.

3 Kana-Kanji Conversion

In this section, we describe the Kana-Kanji conversion model we employed to achieve the consonant-to-Kanji conversion. As we mentioned, the input to the back-end part passed from the front-end part is a lattice of possible consonant sequences. We therefore have to "guess" the possibly omitted vowels somehow and convert the sequences back into intended Kanji sequences. However, it would be an exponentially large number if we expand the input consonant sequence to all possible Kana se-

[2] http://en.wikipedia.org/wiki/Graffiti_2

quences. Therefore, instead of attempting to restore all possible Kana sequences, we directly “decode” the input consonant sequence to obtain the Kanji sequence. We call this process *direct consonant decoding*, shown in Figure 3. It is basically the same as the vanilla Viterbi search often used for Japanese morphological analysis (Kudo et al., 2004), except that it runs on a consonant sequence. The key change to this Viterbi search is to make it possible to look up the dictionary directly by consonant substrings. To do this, we convert dictionary entries to possible consonant sequences referring to Microsoft IME Kana Table[3] when the dictionary structure is loaded onto the memory. For example, for a dictionary entry 福袋/フクブクロ *hukubukuro*, possible consonant sequences such as “hkbkr,” “hukbkr,” “hkubkr,” “hukubkr,” “hkbukr,”... are stored in the index structure.

As for the conversion model, we employed the discriminative Kana-Kanji conversion model by Tokunaga (2011). The basic algorithm is the same except that the Viterbi search also runs on consonant sequences rather than Kana ones. We used surface unigram, surface + class (PoS tags) unigram, surface + reading unigram, class bigram, surface bigram as features. The red lines in Figure 3 illustrate the finally chosen path.

The suggestion candidates, which is to show candidates such as *hukubukuro* (lucky bag) and *hontai* (body) for an input “H,” are chosen from 2000 most frequent query fragments issued in 2011 at Rakuten Ichiba[4]. We annotate each query with Kana pronunciation, which is converted into possible consonant sequence as in the previous section. At run-time, prefix search is perfomed on this consonant trie to obtain the candidate list. The candidate list is sorted by the frequency, and shown to the user supplementing the Kana-Kanji conversion results.

4 Experiments

In this section, we compare KooSHO with *Siri* and a software keyboard system. We used the following three training corpora: 1) BCCWJ-CORE (60,374 sentences and 1,286,899 tokens)[5], 2) EC corpus, consists of 1,230 product titles and descriptions randomly sampled from Rakuten Ichiba (118,355 tokens). 3) EC query log (2000 most frequent query fragments issued in 2011 at Rakuten Ichiba) As the dictionary, we used UniDic[6].

Character Recognition Firstly, we evaluate the accuracy of the character recognition model. For each letter from “A” to “Z,” two subjects attempted to type the letter for three times, and the accuracy how many times the character was correctly recognized was measured.

We observed recognition accuracy varies from letter to letter. Letters which have similar forms, such as “F” and “T” can be easily misrecognized, leading lower accuracy. For some of the cases where the letter shape completely contains a shape of the other, e.g., “F” and “E,” recognition error is inevitable. The overall character recognition accuracy was 0.76.

Kana-Kanji Conversion Secondly, we evaluate the accuracy of the Kana-Kanji conversion algorithm. We used ACC (averaged Top-1 accuracy), MFS (mean F score), and MRR (mean reciprocal rank) as evaluation measures (Li et al., 2009). We used a test set consisting of 100 words and phrases which are randomly extracted from Rakuten Ichiba, popular products and query logs. The result was ACC = 0.24, MFS = 0.50, and MRR = 0.30, which suggests the right choice comes at the top 24 percent of the time, about half (50%) the characters of the top candidate match the answer, and the average position of the answer is 1 / MRR = 3.3. Notice that this is a very strict evaluation since it does not allow partial input. For example, even if “フィットネスシューズ” *fittonesushu-zu* (fitness shoes) does not come up at the top, one could obtain the same result by inputting “フィットネス” (fitness) and “シューズ” (shoes) separately. Also, we found that some spelling variations such as まつげ and まつ毛 (both meaning eyelashes) lower the evaluation result, even though

[3] http://support.microsoft.com/kb/883232/ja

[4] http://www.rakuten.co.jp/

[5] http://www.ninjal.ac.jp/corpus_center/bccwj/

[6] http://www.tokuteicorpus.jp/dist/

they are not a problem in practice.

Overall Evaluation Lastly, we evaluate the overall input accuracy, speed, and user experience comparing *Siri*, a screen keyboard (Tablet PC Input Panel) controlled by Kinect using KinEmote[7], and KooSHO.

First, we measured the recognition accuracy of *Siri* based on the 100 test queries. The accuracy turned out to be 85%, and the queries were recognized within three to four seconds. However, about 14% of the queries cannot be recognized even after many attempts. There are especially two types of queries where voice recognition performed poorly — the first one is relatively new, unknown words such as オーガランド (ogaland), which obviously depends on the recognition system's language models and the vocabulary set. The second the is homonyms, i.e., voice recognition is, in principle, unable to discern multiple words with the same pronunciation, such as "包装" (package) and "放送" (broadcast) *housou*, and "ミョウバン" (alum) and "明晩" (tomorrow evening) *myouban*. This is where KooSHO-like visual feedback on the conversion results has a clear advantage.

Second, we tried the screen keyboard controlled by Kinect. Using a screen keyboard was extremely difficult, almost impossible, since it requires fine movement of hands in order to place the cursor over the desired keys. Therefore, only the time required to place the cursor on the desired keys in order was measured. The fact that users have to type out all the characters including vowels is making the matter worse. This is also where KooSHO excels.

Finally, we measured the time taken for KooSHO to complete each query. The result varied depending on query, but the ones which contain characters with low recognition accuracy such as "C" (e.g., "チーズ" (cheese)) took longer. The average was 35 seconds.

Conclusion and Future Works

In this paper, we proposed a novel environment for Japanese text input based on aerial hand gestures called KooSHO, which provides an integrated experience of character input, Kana-Kanji conversion, and search result visualization. This is the first to propose a Japanese text input system beyond single characters based on hand gestures. The system has several novelties, including 1) non-restrictive character forms, 2) robust continuous recognition and Kana-Kanji conversion, and 3) faster input by suggestions and consonant input. The comparison with voice recognition and a screen keyboard showed KooSHO can be a more practical solution compared to the screen keyboard.

Since KooSHO is an integrated Japanese input environment, not just a character recognition software, many features implemented in modern input methods, such as fuzzy matching and user personalization, can also be implemented. In particular, how to let the user modify the mistaken input is a great challenge.

References

Xinying Han, Hiroaki Seki, Yoshitsugu kamiya, and Masatoshi Hikizu. 2007. Wearable handwriting input device using magnetic field. In *Proc. of SICE*, pages 365–368.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proc. of EMNLP*, pages 230–237.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of news 2009 machine transliteration shared task. In *Proc. of NEWS*, pages 1–18.

Tomonari Sonoda and Yoishic Muraoka. 2003. A letter input system of handwriting gesture (in Japanese). *The Transactions of the Institute of Electronics, Information and Communication Engineers D-II*, J86-D-II:1015–1025.

Kumiko Tanaka-Ishii, Yusuke Inutsuka, and Masato Takeichi. 2001. Japanese text input system with digits. In *Proc. of HLT*, pages 1–8.

Hiroyuki Tokunaga, Daisuke Okanohara, and Shinsuke Mori. 2011. Discriminative method for Japanese kana-kanji input method. In *Proc. of WTIM*.

Ho-Sub Yoon, Jung Soh, Byung-Woo Min, and Hyun Seung Yang. 1999. Recognition of alphabetical hand gestures using hidden markov model. *IEICE Trans. Fundamentals*, E82-A(7):1358–1366.

[7] http://www.kinemote.net/