SEMI-SYNCHRONOUS SPEECH AND PEN INPUT

Yasushi Watanabe, Kenji Iwata, Ryuta Nakagawa†, Koichi Shinoda, and Sadaoki Furui

Tokyo Institute of Technology, Department of Computer Science
{yasusi, iwata}@ks.cs.titech.ac.jp, {shinoda, furui}@cs.titech.ac.jp
†Nagasaki University, Graduate School of Biomedical Sciences
rnkgw@nagasaki-u.ac.jp

ABSTRACT

This paper proposes a new interface method using semi-synchronous speech and pen input for mobile environments. In this interface, a user speaks while writing, where pen input complements speech to achieve higher recognition performance than speech alone. A multimodal recognition algorithm that can handle the asynchronisity of the two modes using a segment-based unification scheme is proposed. This method is evaluated under noisy conditions with four different pen-input interfaces: character, stroke, pen-touch, and point-to-character, each of which is assumed to be given for a phrase unit in speech. It is confirmed that the recognition accuracy is improved by the proposed method in comparison with that by speech alone in all the pen-input conditions.

Index Terms— User interfaces, speech recognition, handwritten character recognition

1. INTRODUCTION

Mobile devices such as PDAs and cellular phones are widely used. An interface in which users can easily input long sentences is desirable for e-mailing, scheduling, and other purposes. At present, the ten-key pad, speech, or handwriting are used for this purpose. A user often has difficulty entering long sentences with the ten-key pad. While a speech interface has a recognition accuracy of more than 90% under quiet conditions and its input speed is faster than by key, its performance is seriously degraded in a noisy, mobile environment. On the other hand, the recognition accuracy of handwritten characters is generally high and not influenced by noise. However, its input speed is much slower than speech. Thus, speech and pen inputs have complementary merits and demerits, which led us to combine them. In this paper, we propose a multimodal interface using semi-synchronous input of the two modes for entering long sentences. In this interface, a user speaks while writing or writes while speaking, where the two modes complement one another to improve the recognition performance. This interface is expected to be more robust against noise than speech alone, and its input speed is faster than by pen alone.

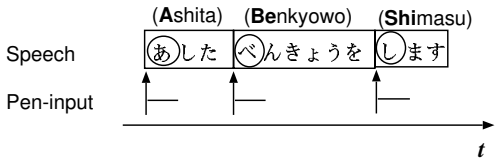
Since the asynchronisity of the two modes occurs in practice, an algorithm that can handle this should be provided. There are two main types of multimodal recognition methods: one integrates signals at the feature level (*feature fusion*) and the other at the semantic level (*decision fusion*) [1]. Feature fusion combines multiple input streams in a single feature space. The correlation structure between the two modes can be taken into account automatically via learning. Feature fusion is generally appropriate for closely coupled and synchronized modes, such as speech and lip movements [2]. However, continuous speech and pen inputs often differ in their input speeds and beginning times. For these semi-synchronous inputs, integrating modes at the feature level is difficult. On the other hand, decision fusion generally includes an individual recognizer for each mode and combines the output of the recognizers after the input finishes. These individual recognizers can be trained using unimodal data, which are relatively easy to collect or are already publicly available for modes like speech and handwriting [1]. However, decision fusion cannot fully utilize the correlated information of multiple modes along the time axis. Furthermore, determining when to combine multiple modes is difficult when one or more of the input modes is a sequence of segments whose boundaries are unknown. The typical example for this case is continuous speech with many words. Thus, neither feature nor decision fusion is clearly applicable for our interface.

We propose a multimodal recognition algorithm for semi-synchronous speech and pen input. In this algorithm, the input for each mode is divided into segments, and corresponding segments in the two modes are combined to produce recognition results. The algorithm can handle segment-length asynchronisity of the two modes. There have been several related studies with similar motivation. For example, Ban *et al.*[3] proposed a speech interface with finger-tapping at word boundaries. Zhou *et al.*[4] combined speech and pen input for entering a Chinese character. Hui *et al.*[5] used speech and pen gestures for navigational inquiries. While all of them were proven effective, they had limitations either in speech input information or in its application. Different from these works, we use pen input to improve recognition performance for general *large vocabulary continuous speech recognition* (LVCSR). In addition, we developed an algorithm that combines speech and pen input in a segment-based unification scheme. We evaluated four different pen-input interfaces, in which a pen-input is given corresponding to each phrase in continuous speech.

In Section 2, we describe four different pen-input interfaces. In Section 3, we describe our multimodal recognition algorithm. In Section 4, the method of the experiments and our results are shown.

2. INTERFACE

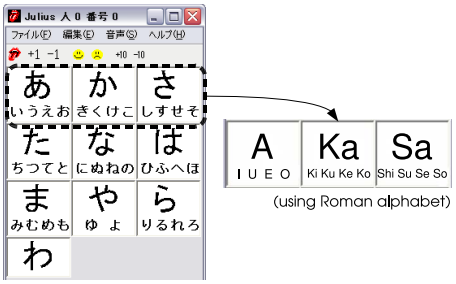
We propose an interface using semi-synchronous speech and pen input for entering Japanese sentences that consist of many phrases. Inputting the same information by speech and pen is practically impossible since the speed of pen input is very slow compared with speech. Therefore, we need to make constraints on pen input that corresponds to each speech phrase. First, we assume that a pen-input will be given at the beginning of a speech phrase (Fig. 1(a)).



(a) Relationship between speech and pen input. Arrow indicates beginning time of pen-input for character surrounded by circle. Each box in speech indicates one phrase.

(b) Example of character inputs for sentence in (a). Here, beginning character for each phrase is input.

(c) Example of stroke inputs for sentence in (a). Here, first stroke of beginning character for each phrase is input.



(d) Point-to-character interface. Each region corresponds to 1-5 characters that share same consonant.

Fig. 1. Examples of interfaces

Second, we assume that a user will try to synchronize the beginning of a pen-input with the beginning of the corresponding speech phrase. Here, a phrase means Japanese *bunsetsu*, which contains one or more words. A user would not need to give pen-inputs for all phrases or insert pauses between phrases. We propose the following four interfaces based on these assumptions.

1. **Character input** A user writes the initial character of each phrase in *hiragana*, phonetic characters for Japanese. The interface has several input boxes, and a user writes one character in one box from left to right (Fig. 1(b)).
2. **Stroke input** This interface is similar to the previous interface. A user writes only the initial stroke of the initial character instead of all of the strokes of the actual character (Fig. 1(c)).
3. **Pen-touch** A user inputs a pen touch at the beginning of each phrase instead of writing characters, which is similar to the finger-tapping interface [3].
4. **Point-to-character** Each *hiragana* is either a vowel (V) or a consonant followed by a vowel (CV). They are classified into 10 groups: one corresponds to the characters only for vowels and the rest correspond to the characters sharing the same consonant. There are five vowels and nine consonants. With this classification, this interface provides a character table (Fig. 1(d)), each element of which indicates a character group. A user taps the group to which the first character of the speech phrase belongs.

Table 1 summarizes these four interfaces from the viewpoint of usability. Regarding the amount of time a pen-input takes, pen-touch(3) and point-to-character(4) are the shortest and character-input(1) is the longest. Regarding the amount of information a pen-input gives, character-input(1) is the best and pen-touch(3) is the worst. Pen-touch(3) is the easiest to synchronize with speech and character-input(1) is the most difficult. We report our evaluation of these interfaces in Section 4. Although these interfaces may seem difficult to use, users can use them without difficulty after approximately five minutes of practice.

Table 1. Summary of interfaces from viewpoint of usability

Interface	Character	Stroke	Pen-touch	Point
Time pen input takes	Poor	Fair	Good	Good
Information pen input gives	Good	Fair	Poor	Fair
Synchronous ease	Poor	Fair	Good	Fair

3. MULTIMODAL RECOGNITION

The multimodal recognition algorithm we developed for the interface uses a two-pass search process. This algorithm is based on the conventional LVCSR algorithm, where the linguistic unit is a word. In the first pass, speech and pen input recognition hypotheses are merged online to produce a word graph. In the second pass, each hypothesis in the word graph is rescored using information about the time-lag distribution between speech and pen inputs.

3.1. First pass

A user will try to synchronize the beginning time of a pen-input with that of its corresponding speech phrase, but asynchronisity of the two modes always occurs in practice; some users start a pen-input before the corresponding speech phrase starts, and others start after the corresponding speech phrase started. To merge pen input recognition hypotheses with their corresponding speech recognition hypotheses, this asynchronisity needs to be taken into consideration. We assume that each user has his or her own tendency for time-lags between the two modes. Let a pen input beginning time be t and the corresponding phrase beginning time be t'. The time-lag Δt is defined as $t - t'$. The mean μ of Δt is estimated by using the user's data. In the recognition phase, the pen-input beginning time t is corrected to $t - \mu$. The mean μ is different among users and can also be a negative value. In the first pass, each speech recognition hypothesis is merged with the pen input recognition hypothesis weighted by factor α at the time the pen-input begins. To achieve this, the speech recognition process suspends from the time the pen-input begins to

the time the pen input recognition finishes. This algorithm is shown below.

Step 0: Set time $t = 0$.
Step 1: If the beginning of pen-input is detected, go to Step 4.
Step 2: If speech input ends, terminate.
Step 3: Set $t = t + 1$, and go to Step 1.
Step 4: Operate the following for each hypothesis h of speech recognition at time t.

1. Extract a initial character C of a speech recognition hypothesis.
2. Obtain the pen input recognition likelihood of C, $L(C)$.
3. Calculate the likelihood for h as follows

$$L(h) = L_s(h) + \alpha L(C) \ .$$

Here, $L_s(h)$ is speech recognition likelihood of h, and α is a weighting factor.

Step 5: Set $t = t + 1$, and go to Step 1.

The hypotheses having a word whose initial character has high probability in the pen input recognition are likely to remain within the beamwidth, and accordingly the candidates are narrowed down effectively.

3.2. Second pass

While we consider the mean μ of a time-lag between the two modes in the first pass, its variance is also important for our multimodal recognition. A phrase hypothesis whose beginning time is *closer* to the pen-input beginning time should be more probable than a hypothesis whose beginning time is *far-off*. In the second pass, we assume that the time-lag between speech and pen inputs follows a normal distribution, and each hypothesis in the word graph is rescored using pen input recognition hypotheses weighted by a coefficient that is a function of its probability density.

Let a pen-input be c_n and the first frame of c_n be i_n. Let the first frame of a hypothesis in the word graph be i. The difference between i_n and i is defined as δ, which is assumed to follow normal distribution $p(\delta)$ with variance σ^2. Then, a weighting factor corresponding to the hypothesis is defined as $\beta p(\delta)$. Each hypothesis of which the first frame exists within I frames before and after the pen-input is merged with a pen input recognition hypothesis weighted by $\beta p(\delta)$. Here, β is a control parameter of a weighting factor. Let the likelihood for a hypothesis h in the word graph be $L(h)$ and the pen input recognition likelihood be L_p. Then, the likelihood for h is rescored as

$$L'(h) = L(h) + \beta p(\delta) L_p \ . \qquad (1)$$

Figure 2 shows an example of this rescoring. This process is repeated for every pen-input $n = 1, \cdots N$.

4. EXPERIMENT

4.1. Experimental conditions

We collected simultaneous input data of speech and pen from 10 Japanese male subjects in an office environment. Each subject input 20 sentences for each of the four interfaces in Section 2. The 20 sentences consisted of 5 sentences chosen freely by each subject and 15 sentences chosen randomly from the ASJ-PB database with a phonetically-balanced sentence set and the ASJ-JNAS database with a sentence set from Japanese newspaper articles. We added exhibition hall noise in the JEIDA noise database [6] to these 800 sentences (20 sentences $\times$ 4 interfaces $\times$ 10 subjects) at 20, 15, and 10 dB SNR

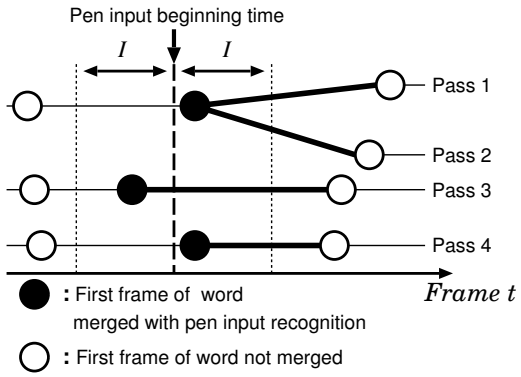


Fig. 2. Example of word graph and pen input. Hypotheses merged with pen input recognition hypotheses are expressed in thick lines.

and used a total of 3,200 sentences involving 800 sentences in clean condition. The subjects used each interface for about 10 minutes to get accustomed to it before our evaluation.

We used triphone HMMs with 16 mixture components per state included in the IPA Japanese Dictation Toolkit (1999 version) [7] as the acoustic model. The word dictionary was created from articles of the *Mainichi* newspaper from 1995 to 2001. We removed symbols without pronunciation from the articles and extracted 60,000 words with high frequency.

For the handwritten character recognition, we used continuous HMMs trained on 43,800 characters, which were *hiragana* by 10 writers obtained from the online handwritten characters database [8]. Our handwritten character recognition was based on stroke-based left-to-right HMMs [9], where a recognition unit was a stroke and the number of stroke units was 25. Pen-down strokes had three states and pen-up strokes had one state without self-loop probability. The number of mixture components for each state was one. We created a handwritten character dictionary that consists of 71 *hiragana*, each of which was represented as a concatenation of strokes. Characters for which the writing order was significantly different among writers had multiple entries in the dictionary. As a result, the number of handwritten characters in the dictionary was increased to 82.

We implemented our algorithm with the speech decoder Julius [10]. We obtained the pen input recognition likelihood in advance using Julius and unified the two modes offline. The mean μ of the time-lag of speech and pen inputs was estimated in advance using the test set for each subject. The values for the weighting factor α in the first pass, β in the second pass, the variance σ^2 of time-lag, and the window size I were the same for all the subjects optimized using the test data of all the subjects for each condition.

4.2. Results

Table 2 shows the experimental results of speech recognition and our proposed multimodal recognition with speech and pen inputs. In all cases, the recognition accuracies of speech and pen input recognition were higher than those of speech recognition alone. The accuracies of speech recognition among the four interfaces were different because we collected a database of speech and pen inputs for each interface. Comparing the relative word error rate reduction among the four interfaces, point-to-character was the highest and character-input was the second. The interfaces are effective under both clean and noisy conditions.

Table 2. Word accuracies of speech recognition and multimodal recognition with speech and pen inputs (%). ERR shows the relative word error rate reduction.

SNR	Interface	Character	Stroke	Pen-touch	Point
Clean	speech	83.6	84.0	84.5	82.6
	speech+pen	84.6	84.4	84.9	84.5
	ERR	6.1	2.5	2.6	10.9
20 dB	speech	76.8	76.7	79.2	76.5
	speech+pen	78.5	77.1	80.0	79.0
	ERR	7.3	1.7	3.8	10.6
15 dB	speech	67.1	63.4	66.8	66.8
	speech+pen	68.3	64.0	67.7	69.7
	ERR	3.6	1.6	2.7	8.7
10 dB	speech	41.6	41.0	43.0	42.0
	speech+pen	44.4	41.8	43.8	46.0
	ERR	4.8	1.4	1.4	6.9

Table 3. Average rank of hypotheses whose initial character is correct.

Interface	Character	Stroke	Point
Speech	634.7	622.0	614.0
Speech+pen	499.4	619.3	229.5

Next, we investigated how efficiently the proposed method narrowed the search space in the first pass. We examined the rank of each hypothesis whose initial character was correct at the frame when a pen input started. Table 3 shows the ranks of such hypotheses averaged over all the pen inputs in the data. The ranks of those hypotheses were improved in all the proposed interfaces. Point-to-character interface, which had the highest recognition performance, also had the highest improvement.

Table 4 shows the mean and variance of the time-lag between speech and pen inputs and the average pen-input frequency in one sentence. Regarding the time-lag variance, character-input was the highest. This result shows that synchronizing pen input with speech was most difficult when using this interface. Regarding the pen-input frequency, pen-touch was the highest. This result indicates that pen-touch takes the shortest amount of input time. Point-to-character had the highest recognition performance. This result indicates that the asynchronisity of the two modes influences recognition performance, since point-to-character has a significantly smaller time-lag variance than other interfaces.

To evaluate the usability of the interfaces, we asked each subject to rank the four interfaces, permitting more than one interface to have the same rank. Table 5 shows the average rank for each interface across all subjects. Pen-touch had the highest rank, and the other interfaces had approximately the same rank. The easiest interface differed among users. Pen-touch was the easiest on average, but its relative word error rate reduction was lower. This result indicates that the most appropriate interface will vary by user and environment.

5. CONCLUSION

We proposed an interface using semi-synchronous speech and pen input and an algorithm using a segment-based unification scheme to handle the asynchronisity of the two modes . We evaluated four different pen-input interfaces. By combining pen input with continuous speech, recognition accuracy was improved beyond speech alone for all interfaces. The interfaces are effective under both clean and noisy conditions. In addition, choosing an appropriate interface for the users and their environments was demonstrated to be important. Our future work includes: 1) adapting model parameters in multimodal recognition to users and their environments, 2) implementing an on-line algorithm that recognizes speech and pen input in real time, and 3) extending our work to different languages and modes.

Table 4. Time-lag of speech and pen inputs (frame)

Interface	Character	Stroke	Pen-touch	Point
Mean	-0.7	0.8	3.5	5.6
Standard variation	16.2	11.2	8.7	9.7
Average pen-input frequency in one sentence	3.3	4.4	4.8	4.6

Table 5. Average rank of ease of use for all subjects

Interface	Character	Stroke	Pen-touch	Point
Average rank	2.8	3.1	1.0	2.7

6. ACKNOWLEDGMENTS

This research was partially supported by JSPS Grants-in-Aid for Scientific Research (B) 15300054. Our thanks go to Nakagawa Laboratory of Tokyo University of Agriculture and Technology for providing the online handwritten characters database.

7. REFERENCES

[1] L. Wu, S. L. Oviatt, and P. R. Cohen, "Multimodal Integration-A Statistical View," *IEEE Trans. Multimedia*, vol. 1, no. 4, pp. 334–341, 1999.

[2] S. Tamura, K. Iwano, and S. Furui, "Multi-modal speech recognition using optical-flow analysis for lip images," *The Journal of VLSI Signal Processing - Systems for Signal*, Image, and Video Technology, vol. 36, iss. 2-3, pp. 117–124, 2004.

[3] H. Bann, C. Miyajima, K. Itou, K. Takeda, and F. Itakura, "Speech recognition using synchronization between speech and figure tapping," *Proc. INTERSPEECH 2004 – ICSLP*, Jeju, Korea, 2004.

[4] X. Zhou, Y. Tian, J. Zhou, F. K. Soong, and B. Dai, "Improved Chinese character input by merging speech and handwriting recognition hypotheses," *Proc. ICASSP 2006*, Toulouse, France, vol. 1, pp. 609–612, 2006.

[5] P. Y. Hui and H. M. Meng, "Joint Interpretation of Input Speech and Pen Gestures for Multimodal Human-Computer Interaction," *Proc. INTERSPEECH 2006 – ICSLP*, Pittsburgh, Pennsylvania, pp. 1197–1200, 2006.

[6] S. Itahashi, "A noise database and Japanese common speech data corpus," *J. Acoust. Soc. Jpn.*, vol. 47, no. 12, pp. 951–953, 1991 (in Japanese).

[7] http://winnie.kuis.kyoto-u.ac.jp/dictation/

[8] M. Nakagawa, T. Higashiyama, Y. Yamanaka, S. Sawada, L.V. Tu, and K. Akiyama, "Collection and Utilization of On-line Handwritten Character Patterns Sampled in a Sequence of Sentences without Any Writing Instructions," Technical Report of IEICE, 95, 278, pp. 43–48, 1995 (in Japanese).

[9] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke Approach to HMM-based On-line Kanji Handwriting Recognition," *Proc. of ICDAR 2001*, pp. 491–495, 2001.

[10] A. Lee, T. Kawahara, and K. Shikano, "Julius – an open source real-time large vocabulary recognition engine," *Proc. EUROSPEECH 2001*, pp. 1691–1694, 2001.