Speech Pen: Predictive Handwriting based on Ambient Multimodal Recognition

Kazutaka Kurihara
Dept. of Computer Science,
The Univ. of Tokyo
Tokyo, 1130033, Japan
qurihara@nifty.com

Masataka Goto
National Institute of Advanced Industrial Sci. and Tech. (AIST)
Ibaraki, 3058568, Japan
m.goto@aist.go.jp

Jun Ogata
National Institute of Advanced Industrial Sci. and Tech. (AIST)
Ibaraki, 3058568, Japan
jun.ogata@aist.go.jp

Takeo Igarashi
Dept. of Computer Science, The Univ. of Tokyo / JST PRESTO
Tokyo, 1130033, Japan
takeo@acm.org

ABSTRACT

It is tedious to handwrite long passages of text by hand. To make this process more efficient, we propose predictive handwriting that provides input predictions when the user writes by hand. A predictive handwriting system presents possible next words as a list and allows the user to select one to skip manual writing. Since it is not clear if people are willing to use prediction, we first run a user study to compare handwriting and selecting from the list. The result shows that, in Japanese, people prefer to select, especially when the expected performance gain from using selection is large. Based on these observations, we designed a multimodal input system, called speech-pen, that assists digital writing during lectures or presentations with background speech and handwriting recognition. The system recognizes speech and handwriting in the background and provides the instructor with predictions for further writing. The speech-pen system also allows the sharing of context information for predictions among the instructor and the audience; the result of the instructor's speech recognition is sent to the audience to support their own note-taking. Our preliminary study shows the effectiveness of this system and the implications for further improvements.

Author Keywords

Predictive handwriting, speech recognition, handwriting recognition, multimodal interface, context-sharing, education, presentation tool.

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies.

INTRODUCTION

Lecturing and note-taking is one of mankind's fundamental communication and information processing techniques. It is also a good example of multimodal interactions in which an instructor and the audience communicate with each other by speech, body gestures, and utilizing written materials naturally and effectively.

With advances in digital technologies, many systems have been designed to support instructors and the audience during lectures. Some systems focus on annotating pre-authored slides [1,3,19,31] and some systems are primarily designed for writing from scratch [8,9,30,33]. Writing is superior to just showing pre-authored slides in that the presentation becomes more flexible and more engaging [32]. In addition it saves the time that would be required to prepare complete slides.

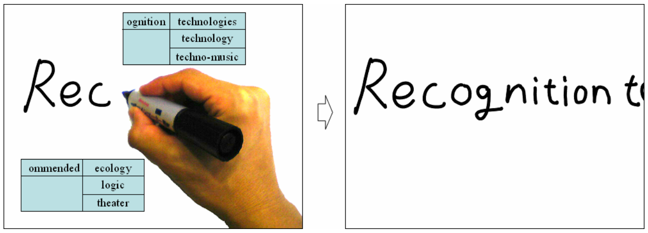

Figure 1: Digital Writing supported by Ambient Multimodal Recognition. The system shows multiple predictions around the pen (left). The user can select one and paste it in the board to save tedious manual writing (right).

One problem with writing is that it is tedious to write long texts by hand. It is reported that as much as 18% of lecture time is consumed by writing on the board [17]. Although it is not always desirable to reduce the time (e.g. it helps students to follow the lecture), excessive writing may distract the writer and the audience.

In this paper, we propose predictive handwriting to reduce the burden of manual writing for the Japanese language. The system predicts possible next words based on speech recognition and handwriting recognition, and allows the user to choose a desired word or sentence from a list to reduce manual writing. Prediction has been frequently used in typed text entry, but we are not aware of a previous system that has incorporated prediction for handwriting.

The biggest concern as to whether such a system will be effective is that users might not prefer predictive methods because of the cognitive overload required to choose the correct prediction [21, 24]. To counter this concern and justify our approach, we first performed a user study that examines the user's behavior in Japanese writing. The result shows that people prefer selecting to writing in general and that selection is especially preferred for words consisting of many strokes.

Based on these observations, we developed a prototype system called speech-pen to examine the possibilities of predictive writing. Figure 1 illustrates the basic concept of the system. This system helps the instructor's manual writing – not the entry of *typed* texts – by suggesting possible further writing based on speech and handwriting recognition (Figure 1 left). If the instructor finds a correct prediction in the list, he can paste it on the board to save manual writing. If not, he can simply ignore the predictions and continue writing. The system uses a customized font that mimics the instructor's own handwriting to seamlessly integrate the manual writing and automatically generated texts (Figure 1 right).

In addition to supporting the instructor's writing, the speech-pen system also supports the audience's note-taking by providing similar predictions. The result of the instructor's speech recognition is sent to each of the audience's tablet PCs and used as a context to generate correct predictions for note-taking. We call this "ambient context" sharing because it is a kind of context-sharing usually done in the background.

BACKGROUND

Digital Writing and Note-Taking

Digital writing for real time presentations has been discussed mainly in the context of electronic whiteboard systems. Early systems, such as Xerox Liveboard [9] and the Tivoli system [30] are mainly designed for small group meetings. They provide various interfaces to organize the instructor's handwriting on the board. Some recent systems are more specifically designed for large-scale classroom presentations [1,3,11,19]. Most systems emphasize the integration of digital writing with pre-authored presentation slides.

Some systems are designed to support note-taking. The Audio Notebook [33], Dynomite [36] and many other systems record the instructor's speech and associate it with the handwritten notes. The user can quickly play the audio track by specifying the corresponding handwritten note. In these cases the speech is not converted into text but is simply recorded as audio data. eClass [1] experimentally uses speech recognition to generate time-stamped transcripts of lectures but it is not yet readable because of its low recognition accuracy.

Livenotes [19] and StuPad [1] allow listeners to share the slides on their tablet PCs and discuss issues with other listeners by collaborative note-taking in the shared space. NotePals [7] provides a browser of uploaded shared handwritten notes. Denoue et al. proposed a system in which users can share their handwritten words as raw images to increase text input speeds in note-taking [8]. One of the problems of these raw handwriting sharing systems is the difficulty of reading and reusing other peoples' handwriting. We pursue similar goals but the shared context generated by speech and handwriting recognition is basically invisible.

User Interfaces for Recognition Technologies

Recognition technology has a long history of research and development. However in spite of vast research efforts, recognition technology has not yet overcome the fundamental problem of recognition errors [29]. Given this observation, researchers have been exploring various user interface techniques to work with error-prone recognition technologies.

Oviatt [29] investigated the possibilities of improvements on recognition technology using multimodal interfaces and proposed the concept of mutual disambiguation that decreases error rates of recognition by complementary use of multiple modalities. Goto et al [14,15] developed several speech-interface functions that use nonverbal information in speech input. Hindus and Schmandt [16] discussed the utilities of unobtrusive capture of voice interactions in everyday work environments. Lyons et al. [23] augmented this idea and proposed dual-purpose speech. They tried to ease the mental resistance of users in using speech recognition by encoding voice commands in computers in socially acceptable conversations. We take a similar approach, but our goal is assisting the user's handwriting and not giving commands to the computers.

Kaiser [18] and Feng [10] explored *typed* text entry methods with multimodal recognitions of speech and pen. We also combine speech and pen but it is designed to reduce the burden of manual writing with minimum overhead. One of the most important features is that it is relatively tolerant with recognition errors because recognition works only as an auxiliary support, not as the main interaction method as per the *typed* text entry.

Predictive Text Entry

When using mobile devices that only have smaller keyboards or stylus pens, text entry is not so easy and fast. The user may have to type many keys to produce a word (except on mini-QWERTY keyboards [5]), or may have to work on tedious handwriting recognition and correction processes. To improve input efficiency, several approaches have been investigated. One is to improve recognition accuracy by designing robust gestures for the alphabet such as Graffiti. Another approach is to provide efficient software keyboards [13]. SHARK [21] is a combination of the above two approaches.

Predictive text entry is yet another solution. The user can select predictions in a list and paste them instead of entering all the characters [6,12,24]. One of the common arguments aimed at predictive text entry of the English language is whether it is really faster or not. In most of the cases just simply finishing typing the words is often still faster than selecting candidates from a list because of the cognitive load necessary to examine the list. However, we can not simply deduce the same for other languages such as Chinese and Japanese. In these Asian languages thousands of characters are used on a daily basis and each character consists of many strokes. A common input method for them consists of two phases. First, the user inputs phonograms (note that this is the only action necessary for typing English and most European languages). Second, the user converts them to ideograms (Kanji, or Chinese characters) by selecting a candidate from a list. The user can not finish entering words without some interactions with the system because in general there are many ideogram sequences whose pronunciations are the same (see [22,24,35] for the details of the difficult nature of entering these languages). The effectiveness of predictive text entry in these Asian languages is demonstrated by the fact that almost all cell phones available in these Asian countries support predictive text-input methods, and users regularly make use of them.

STUDY ON PREDICTIVE HANDWRITING

This paper proposes *predictive handwriting*, which is an extension of predictive text entry to digital writing. In *predictive handwriting*, the user manually writes characters stroke by stroke using a pen and sometimes word predictions are selected (Figure 1). Masui [24] established an effective predictive input method for *typed* text without quantitative justification of its necessity because it is obvious for Asian languages. On the other hand, it is not so obvious whether *predictive handwriting* is actually preferable because the properties of handwriting are different from those of *typed* text entry. This section describes a user study we performed to address this concern and to collect basic data for designing the system.

Our goal here is to investigate the users' behavior towards handwriting and selection in Japanese writing, but this is a little complicated because many parameters are involved such as number of strokes and number of candidates. Therefore, we first propose a simple practical model that incorporates these parameters and establish hypotheses using the model. We then estimate parameters of the model in the study and examine the hypotheses.

Although our current focus is on *predictive handwriting* of the Japanese language, we expect that the result is also applicable to other Asian languages that use complicated characters, such as Chinese.

Models and Hypotheses

We observed that the user tends to keep typing rather than selecting from a list in English text entry. We also observed that the user tends to select words rather than manually typing everything in Chinese and Japanese text entries. From these observations we can imagine that there must be a certain critical point where the user switches the strategies from typing (or writing) to selecting. Here, we propose a practical model for estimating the point at which handwriting is faster than selecting from the list.

In predictive handwriting, input predictions are effective only when the total time cost for selecting predictions is less than for writing. Suppose the system could always provide the correct predictions. In this simplest case, input predictions are time-effective under the following condition:

$$H(n) > S(m)$$

where *H(n)* is an average time for handwriting a word in terms of *n* the number of strokes, and *S(m)* is an average time for selecting a prediction in terms of *m* the number of candidates. However, the real system does not always provide the correct predictions. Then the user is forced to look at the list and confirm that there is no appropriate candidate and return to write manually. The total time for this action is as follows:

$$S'(m) + H(n)$$

where *S'(m)* is the average time for looking at the list of *m* candidates and confirming that there is no appropriate candidate. Given *p,* the probability of whether the appropriate candidate is in the selection list, we obtain the total time cost for the user's selecting decision:

$$pS(m) + (1-p)\{S'(m) + H(n)\}$$

We can conclude that input predictions are time-effective only under the following condition:

$$H(n) > pS(m) + (1-p)\{S'(m) + H(n)\}$$

To further simplify the formula, we approximate *S'(m)* with *2S(m)* because *S(m)* examines half of the list on average while *S'(m)* examines the entire list all the time. Then by subtracting the left side term from the right side terms, we obtain a function *D*:

$$D(n,m,p) = (2-p)S(m) - pH(n)$$

D is a kind of discriminant that tells us the theoretical advantage of handwriting. It is expected to be faster to write when *D* is positive and vice versa.

Now, we pose the following two hypotheses using *D*:

1) Generally the user prefers to select a word in a list to manually writing the entire word in the case of the Japanese language. Namely, selecting tends to occur relatively regardless of *D*.
2) The user sometimes prefers to write manually when the time for writing is estimated to be less than that for selecting. Namely, writing tends to occur in some cases when *D* is positive.

In the user study, we first obtain a simple estimation of *H(n)* and *S(m)* for the calculation of *D*. Then we examine the hypotheses using the *D* values.

Method

19 volunteers (8 males in their late-teens, 6 females in their late-teens, 4 males in their early-twenties, and 1 female in her late-twenties) participated in the study. They have no

relationships with the computer-science field. They were asked to perform the following 3 tasks on a Tablet PC (Fujitsu FMV-Stylistic TB80): (1) Handwriting task, (2) Selection task, and (3) Combined task. (1) and (2) are for estimating the parameters in *H(n)* and *S(m)*. (3) is for observing the user's preference.

(1) Handwriting Task

This task investigates the writing time in terms of total strokes and number of characters. We chose random words whose numbers of strokes are 1, 4, 8, 12, 16, 20, 24, and 28 (this range covers 85% of all the entries) in a Japanese dictionary with 191,154 entries. For each number of strokes, there are many words that consist of different numbers of characters. Thus we chose three random words corresponding to the minimum, midrange and maximum number of characters. The maximum number of characters were limited to less than or equal to 10. In the end we constructed a 22-word test set per participant to write.

During the study, the system presents the words in random order to the participant and the participant writes each word in a designated writing space (Figure 2 Left). The writing space is divided into 1.6*cm* square cells. Cursive writing was not allowed.

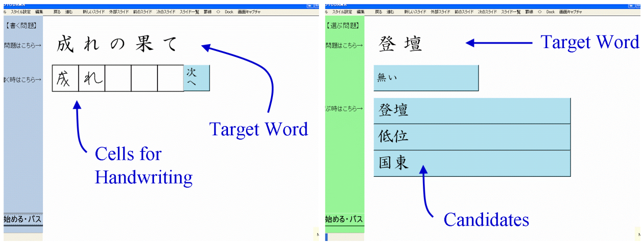

Figure 2: Snapshot of #1 Writing Task (Left) and #2 Selection Task (Right).

(2) Selection Task

This task investigates the selection time in terms of the number of total characters and the number of candidates shown in the list. The number of candidates was 1, 3, or 5. The properties of the word set {number of strokes, number of characters} were the same as task (1). All the false candidates are generated by randomly choosing words whose properties are the same as the target word. In this way we constructed a 66-word (22 × 3) test set per participant.

During the study, participants selected the words in the lists by tapping appropriate candidates. The order of words in the test sets shown was randomized. The space for showing a candidate and selecting was a $1.6cm \times 12cm$ rectangle. Figure 2 (Right) shows a snapshot of the task.

(3) Combined Task

This task investigates the participants' decisions when they are asked to choose writing or selection under various conditions. The combination of {number of strokes, number of characters, number of candidates in candidate list} was the same as task (2). In addition, the probability of whether the appropriate candidate appeared in the selection list was restricted to two simple cases {p=1.0, p=0.5}. The probability p was notified to the participants beforehand to help them establish their strategy. The total test-set consists of 132 (66×2) words per participant. The order of words in the test-sets was randomized.

The system first shows the target word, blank cells for writing the word, and the list of masked candidates (Figure 3). If the user prefers to write the word, he simply starts writing in the cells. If the user prefers to select from a list, he first taps the masked list and the system shows the actual candidates. This allows us to separate the cases where the user decides to write manually without using selection and where the user wanted to select but ended up writing it because the target word was not in the list.

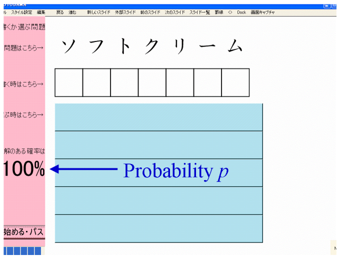

Figure 3: Snapshot of #3 Combined Task.

Result

Figure 4 shows the result of task (1). It was observed that the number of strokes and the time to write words was roughly in proportion. We obtained a simple estimation of *H(n)* using linear regression analysis:

$$H(n) = 0.32n + 0.0831$$

where $R^2 = 0.73$. We omitted the number of characters from this estimation because its effect was negligible.

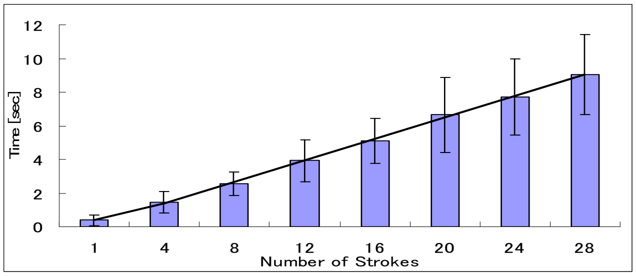

Figure 4: Result of Task (1). Time for writing a word and its number of strokes are roughly in proportion. The black line indicates a linear regression.

Figure 5 shows the result of task (2). It was observed that the constant factor was dominant. We obtained a simple estimation of *S(m)* using linear regression analysis[1]:

[1] Note that Hick's law is not applicable here because the candidates are listed in an unpredictable order.

$$S(m) = 0.105m + 0.6571$$

where $R^2 = 0.16$. This is not a strong fit. However, we need some estimate for selection time to build the system. Thus for engineering purposes this linear regression is adequate to construct an initial system implementation, but we expect that future work can identify a superior model. A reason for the poor fit is large individual differences among users. This can be addressed by adjusting parameters for individual users. As for the number of characters, their effect was negligible and we did not use it in the model.

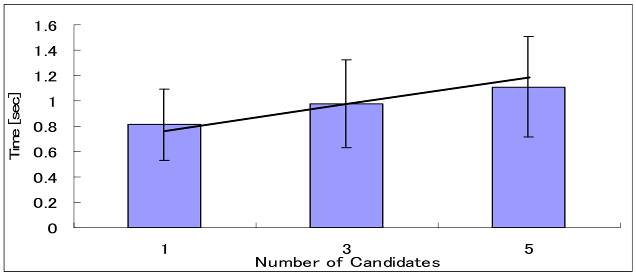

Figure 5: Result of Task (2). Number of candidates in list has a minor effect on the time for selecting a word from the list. The black line indicates a linear regression.

Figure 6 shows the result of task (3) categorized by the *D* value. The histogram above shows the number of the cases when the participants decided to write manually at certain *D*-value condition. The histogram below shows the number of cases of selection. From Figure 6 we observed:

1) Throughout the wide range of the *D* value, the participants decided to select.
2) Handwriting took place only when the *D* value was large. 64% of handwriting took place when the *D* value was zero or positive, and 81% when $D \geq -1$.

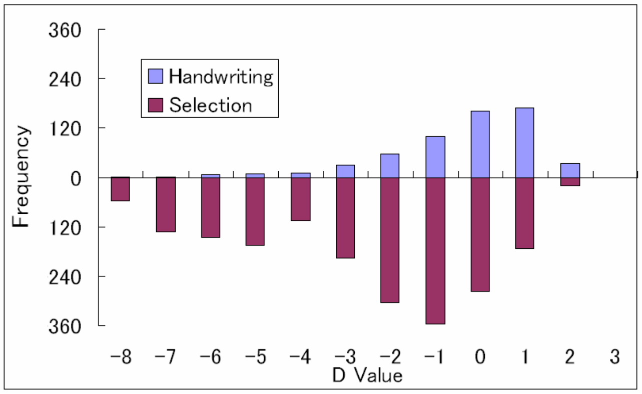

Figure 6: The Distribution of the Participants' Decisions in Various Conditions (the result of task (3)) . The upper half is the number of cases where handwriting was used and the lower half is that where selection was used. *D* is a theoretical metric that estimates the relative advantage of handwriting considering the number of strokes and the number of candidates.

These observations support our hypotheses well. In addition, Figure 6 demonstrates that the quantitative analysis using the *D* value can be used as a rough estimator for the user's behavior.

Findings

Is Predictive Handwriting Effective?

Writing long, complex Japanese words is burdensome and Japanese people are familiar with the process of selecting words from a list. The participants' aggressive tendencies to select reflect this background. However, this result might be slightly biased toward selecting because the participants did not need to compose sentences by themselves in this study. If they actually write while composing sentences in their mind, they might prefer handwriting because it can cause cognitive overhead to examine the list. Future studies should explore this issue further.

From the individual participant's point of view, *D*-value analysis for each participant revealed a diversity of decision strategies. Figure 7 shows the two extreme cases of writing tendencies and selecting tendencies. This result shows that it is important not to force the user to use predictions and to allow both strategies at any time.

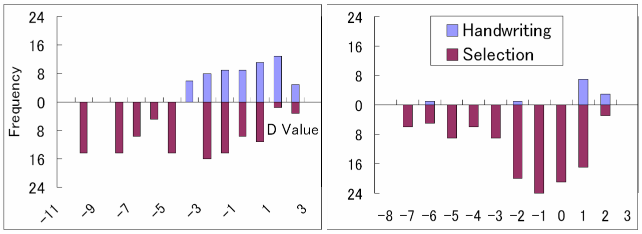

Figure 7: *D*-Value Analyses of Two Extreme Cases. (Left)Writing tendency, (Right) Selecting tendency.

Conservative Predictive Handwriting using the D Value

D-value analysis is useful for suppressing relatively useless input predictions. When the user wants predictions, the system first obtains many candidates from the database based on the user's recent input. Suppose the number of the obtained candidates is *c* and the maximum number of the candidates in the prediction list displayed on a screen is *m'*. At that time the system knows *S(m')* and *H(n)* for each candidate. *p* is roughly estimated[2] by min(*m'*/*c*, 1) for instance. (In the strict sense *p* depends on dictionary adaptations [24] and recognition accuracy if recognitions are involved.)

Finally we obtain estimated *D* values for each candidate. If some of them are positive, the candidates are thought not to be worth providing in the sense of time-efficiency. These candidates can be suppressed for achieving *conservative predictive handwriting*, which will be suitable for the user who prefers the handwriting option.

[2] Note that this calculation is available at anytime during the writing of a word.

THE SPEECH-PEN SYSTEM

The user study in the previous section shows that predictive writing can be useful for the Japanese language. Based on this result, we designed a prototype predictive handwriting tool called “speech-pen” to support digital writing in a class or presentation. It recognizes the instructor’s speech and handwriting and provides predictions for further writing to the instructor. This section describes the details of the system.

System Configuration

The speech-pen system preserves the traditional style of giving lectures and taking notes, but also saves the instructor from manually writing every text by providing predictions for the writing that follows.

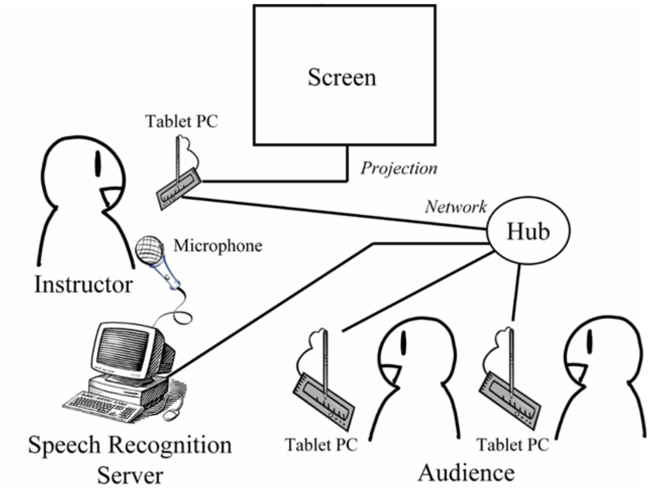

Figure 8: System Configuration.

Figure 8 shows the system configuration of the current speech-pen system. A microphone is attached to the instructor to record his voice. The instructor gives a presentation by writing materials on a large digital surface or tablet PC connected to a projector. The audience also takes notes individually on their tablet PCs. The instructor's speech is recognized by a speech recognition server. The recognition results are distributed to all the users (the instructor and the listeners) over the network. The current prototype supports the Japanese language only while some examples in this paper are in English.

Overview of the User Interface

Figure 9 illustrates how the speech-pen system works from the user’s point of view. Suppose we are in a lecture. The instructor writes on an electric whiteboard while speaking freely (1, 2 in Figure 9). When he pauses writing for a moment or presses a button explicitly, the system displays some predictions that are likely to be written next based on the result of the speech and handwriting recognition. The predictions are placed around his hand so as not to disturb his writing (3 in Figure 9)[3]. The instructor can keep writing when he is not interested in the predictions or when the prediction results are incorrect (4a in Figure 9). If the instructor finds a desired text in the predictions, he can paste it in the board with a single gesture. The text is presented in a font that imitates his own handwriting.

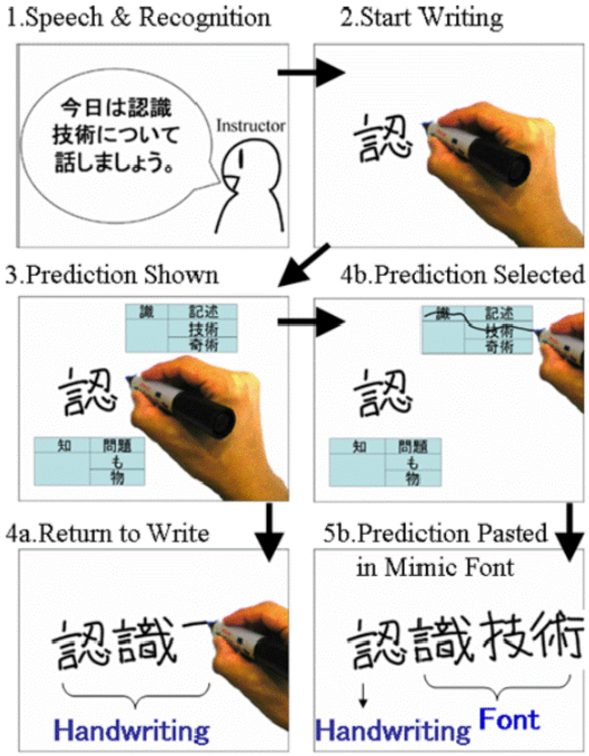

Figure 9: Overview of the User Interface. The instructor says “Today’s topic is about recognition technology,” and writes “Recognition technology” on a white board.

Display of Prediction Result

Multiple prediction results appear around the user’s latest writing (3 in Figure 9). Each prediction result corresponds to the recognition result of an utterance in past speech, or a word in the user’s customized dictionary. The retrieved prediction result is displayed as a collection of multiple sub-candidates (words) as shown in Figure 10. It is a visualization of probabilistic recognition results sorted in the order of likelihood. This interface was originally designed as a method to correct recognition errors for speech-to-text systems [27]. In real-time operation it is superior to zooming interfaces such as [34] in which the user traverses a vast area of candidates.

The latest predictions are shown when the user pauses before starting the next stroke (0.75 seconds in our current prototype). This is empirically designed to detect the

[3] If the instructor is using separate screens for writing and presenting (e.g. tablet PC and projector), these predictions only appear on the writing surface.

natural boundary between individual characters, based on the results of a prior user study. We also decided to always show the latest result of speech recognition at the bottom of the screen because we found that the instructor often writes what he is speaking.

Oviatt [28] reported the existence of individual differences in the order of input modalities when multimodal interfaces are used. Some people tend to speak and write sequentially, and some people prefer to speak and write simultaneously. It might appear that the speech-pen system only supports the speaking-to-writing order. However, the speech-pen system creates predictions using not only sentences that were uttered just before writing but also all sentences that appeared during the current and even past lectures. Important words are often repeated during a lecture and over several lectures. Therefore the speech-pen system provides support for both those who speak and write simultaneously, and those who speak after writing.

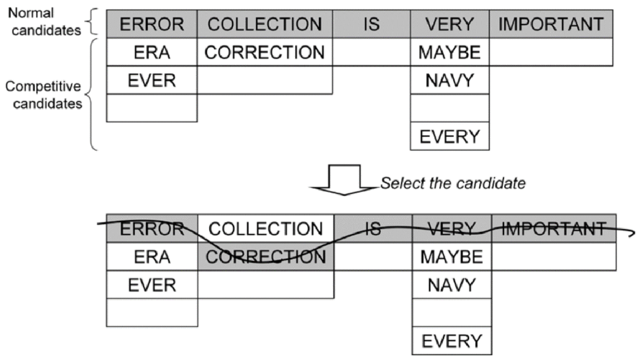

Figure 10: Display of a Prediction Result.

Selecting or Discarding Predictions

When the prediction candidates are displayed, the user can either select a candidate and paste it on the board or discard the candidates and resume writing. The selection is done by crossing [2,4], that is to say, the user draws a stroke over one of the prediction results, tracing the desired words in the list (Figure 10). The selected text is pasted on the board with a font that mimics the user's own handwriting. We currently use a commercial service to generate the customized font [26]. The font size is determined based on the size of recent writing.

The user can simply ignore all predictions and continue with manual writing when they are not useful. As soon as the user starts writing the next character, the prediction disappears. They also disappear when a certain period of time passes after the user finishes writing. Unlike *typed* text entry, digital writing does not require the user to always convert handwriting to *typed* text. Written characters persist as they are and the user can return to writing manually when the predictions are incorrect. This is a significant feature of the speech-pen system that makes it possible to use error-prone speech and handwriting recognition technology in noisy environments.

Sharing Ambient Context

The result of the instructor's speech recognition is distributed to the audience as a shared ambient context. It is used in order to generate prediction results for each member of the audience. The system recognizes a member of the audience's handwriting and retrieves a text in the shared context that begins with the recognized word. As is the case with the instructor, the audience can always ignore the predictions and continue with manual writing. Figure 13 shows an example of writing by an instructor and a member of the audience, obtained in the user study. This result shows that the system successfully supports a variety of individual writing by providing ambient support.

The current prototype system shares speech recognition results only. Our future work is to implement a framework to share other forms of ambient context such as handwriting recognition results. Sharing information on which prediction has been selected by the instructor and the audience would also be useful.

IMPLEMENTATION

This section describes how the speech-pen system generates predictions from speech and handwriting input. The basic idea is to show previous utterances that start with the recently written characters as predictions. While the instructor is speaking, his voice is sent to the speech recognizer and the recognition result is stored in a database. When the instructor starts to write, the handwriting recognizer recognizes the most recent writing. Then the system searches the database using the result of handwriting recognition as a query, and shows the search results to the user as predictions (Figure 11).

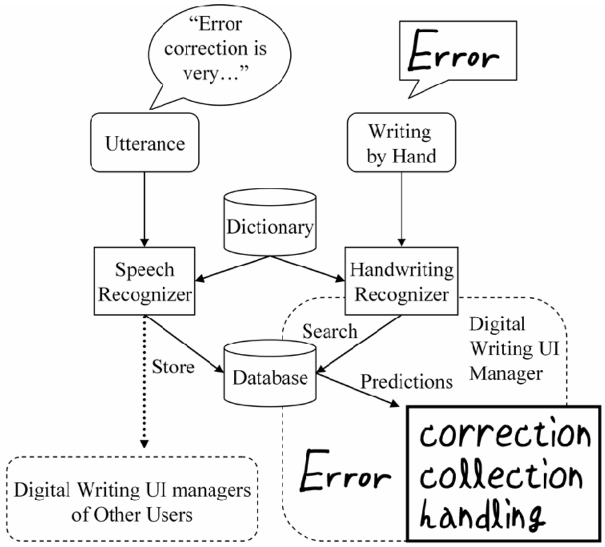

Figure 11: System Architecture of Speech-pen System.

The current implementation is distributed over a LAN for performance reasons. The speech recognizer uses a large-vocabulary continuous speech recognition engine (Julius 3.3 LVCSR [20]) and it runs on a Linux workstation. The handwriting recognizer and the user interface component use the Microsoft Tablet PC platform SDK and run on Tablet PCs.

How Speech-Pen Recognizes Speech

In the speech-pen system, the speech recognizer always works in the background and recognizes the instructor's speech in real time. It first detects the endpoints (beginning and end) of each utterance by using a standard technique that uses short-time energy and recognizes an utterance according to the language model that includes the system vocabulary [27]. Note that even up-to-date HMM-based speech recognizers require a system vocabulary consisting of all the target words. The language model we use is built by learning *Mainichi* newspaper articles, which covered various general topics over a 10 year period. What is important here is that we do not have to register all the terms that will be used in upcoming lectures: even if some terms cannot be recognized, the instructor can simply ignore those wrong predictions. To improve the speech recognition accuracy however, it is recommended that domain specific terms related with upcoming lectures are registered in advance when available. We think it is practical to prepare and register those terms because the instructor usually prepares the contents of lectures beforehand. To prepare the terms, it is also possible to "recycle" speech and handwriting recognition results of the past lectures given by the instructor or other participants. Those terms for the system vocabulary can also be used to improve the handwriting recognition accuracy and be shared by the audience.

The speech recognizer then generates a *confusion network,* which is the result of condensing intermediate hypotheses (a huge internal word graph) of speech recognition. Figure 10 showed a simple example of a graphically represented confusion network. In general the internal word graph itself is too huge for users to understand in the case of large-vocabulary continuous speech recognition. With the confusion network, the user can easily understand competitive candidates (possible alternatives) of recognition results and select the correct word sequence as shown in Figure 10. The details of generating the confusion network and the evaluation of the recognition accuracy are described in [27]. The confusion network is then sent to the database and used as predictions for further writing. The database is distributed to all the digital writing UI managers in the current implementation.

How Speech-Pen Recognizes Handwriting

Our system allows the user to write freely on the blank canvas, i.e. he is not required to write in a cell as seen in many recognition based text entry systems [12]. Therefore, the system first segments the strokes before recognizing them. Figure 12 illustrates an example of a segmentation and recognition results. We segment the strokes into characters and use it as a unit for handwriting recognition. This is because Japanese characters consists of many strokes and can represent a semantic unit. It would be better to use a word as a unit of segmentation for European languages. The result of handwriting recognition is sent to the next step as a sequence of n-best lists. We currently use a recognition engine of the Microsoft Tablet PC Platform SDK [25] and do not consider possible ambiguities in segmentation.

How Speech-Pen Generates Predictions

Given the result of speech recognition and handwriting recognition, the system generates predictions by combining these two. The system first searches the speech recognition results using the handwriting recognition result as a query. Then the result of the search is used for the predictive input suggestion. In the following we describe how to select specific number of predictions by gradually expanding the query.

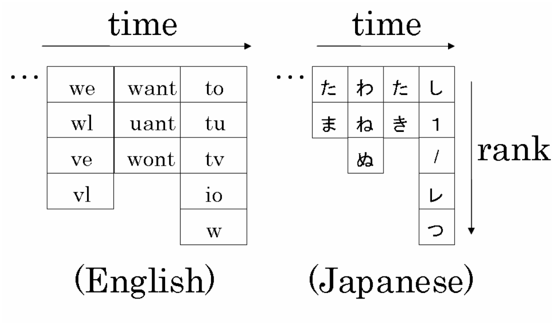

Figure 12: The Result of Handwriting Recognition. It consists of multiple segments and each segment contains multiple (up to five) candidates.

The system first takes the last character (or a word) as a query, and searches for the corresponding character in the confusion networks in the database. If the system returns many matches, the system extends the search query by adding the second last character. That is to say, the system searches for the same two-character sequence in the confusion network, which naturally reduces the number of matching results. The system repeats this procedure until the search returns no result. In the example in Figure 12, the system tries following queries in this order: "to", "want to", "we want to". Finally, the system returns the result that matches the longest query.

It is possible that the search fails at the beginning, i.e. the last character does not appear in the confusion network. In this case the system tries the next best candidate for the last character. If it returns many results, the system extends the query backwards. If it returns no result, it tries the third best candidate and so on. In the example in Figure 12, the system tries "tu" and then "want tu" when "to" returns no result. The search results obtained by the above process are sorted in order of likelihood, considering estimated *D* values. Finally some of the best results are presented to the user (the default is three). This simple algorithm works relatively well in our experience, but there is clearly room for improvement. Our future work will be to investigate various approaches for the search.

EVALUATION OF SPEECH-PEN

Procedure

We performed a preliminary study in order to evaluate the speech-pen system and to obtain the test-users' feedback for further improvements. Eight test-users (male students in their twenties at a university, not majoring CS) participated in the study as volunteers. The task was to write while talking as an instructor and to take notes as a student in a simulated lecture. Each test-user played the role of either an instructor or a student once. We chose "How to cook octopus dumplings (a common Japanese food)" as the topic of the simulated lectures. Our speech recognition engine was not customized for this specific topic. We used an acoustic model and vocabulary that was built from canonical speech of newscasters and not designed for informal conversations. We decided not to optimize the system for this specific test in order to show that our system is still useful with error-prone recognition. Each session took approximately 10 minutes.

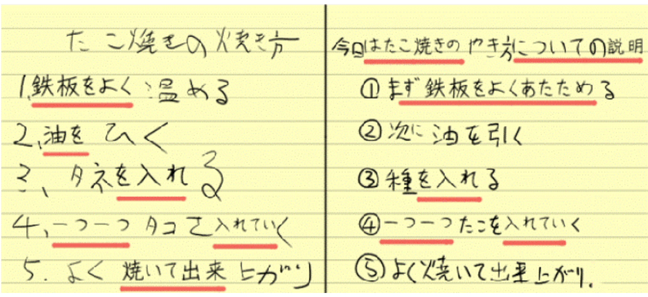

Figure 13: Example Notes Obtained in the Study. The left note is by an instructor and the right note is by a student. A red underline indicates a place where a prediction was used.

General Observations

Figure 13 shows example writing obtained in the study. We added underlines after the study to highlight texts added by the system. We observed a wide diversity of natural-looking writing styles, which signifies a flexibility of our system not seen in other context sharing systems [1,7,19].

Support Ratio

We propose *support ratio* R_{Sup} as a tool to analyze the extent to which the users are supported by the speech-pen system. It is given by the following formula:

$$R_{Sup} = \frac{H(n_{Sup})}{H(n_{All})} \cong \frac{n_{Sup}}{n_{All}}$$

where n_{Sup} is the number of strokes generated by the system[4] and n_{All} is the number of all strokes. The support ratio becomes 0 when all the strokes are written by hand and becomes 1 when all the strokes are generated by the system. It is not our goal to obtain a perfect support ratio—the user basically writes manually and only occasionally uses the predictions. Note that strokes other than text, such as bullets, marks, and drawings[5] always drop the support ratio. This definition is of course a rough approximation to measure the degree of the system's support as a first step. It is more accurate to consider the cognitive load of the user, which is relatively difficult to measure in a natural environment.

Figure 14 shows the support ratios of all test-users labeled A, B, C, D, E, F, G and H. Despite short training, the test-users benefited from the system's support to some extent (22% ~ 70%). We had expected that the support ratios for instructors would be lower than those of students because it might be difficult to write and speak at the same time. However we did not observe such a significant tendency from the result of this small study. In general, novelty effects might bias the test-users' behaviors toward using the predictions aggressively. We need further detailed investigations for obtaining data in more natural setting.

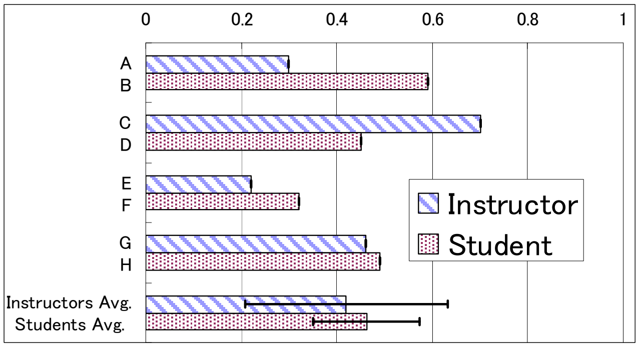

Figure 14: Support Ratios. Eight test-users played either the role of an instructor or a student. The two bars at the bottom show the averages of the instructors' and the students' results.

Feedback from Test-Users

We interviewed the test-users after the test. We first asked for the general impressions of the system. All eight test-users answered that they had positive impressions of the system. They especially found it attractive that the user can use the system only when he wants to, without being forced to do so. We then asked them to compare push-to-talk recognition [23] and background recognition. Six test-users preferred background recognition saying that explicit pushing is tedious, while two preferred the push-to-talk interface saying that recognizing all speech is wasteful because it contains stuttering and irrelevant remarks. We finally asked them to give suggestions to improve the system and obtained comments such as the following: (1) The location to show prediction results needs to be improved. (2) The selection candidates in each prediction result were too small.

CONCLUSION

This paper introduced predictive handwriting as a mean to facilitate the manual writing process on electronic boards. We first showed that predictive writing can be effective at least under certain conditions such as writing long, complex Japanese text. We then introduced a predictive-handwriting system called speech-pen which helps users to write by hand during presentations and lectures using speech recognition and handwriting recognition. The system also allows a sharing of information for *predictive handwriting* among the instructor and the audience in the form of an ambient context. A preliminary study showed the effectiveness of the system and we obtained the users' comments for further improvements about the UI design.

This paper only introduced the basic concept and it requires

[4] Machine-generated text does not actually consist of strokes. We counted the number of strokes necessary to replace them with manual writing.

[5] Microsoft Tablet PC Platform SDK has a function to recognize whether what the user writes is text or drawings. It can be useful for deciding whether to show predictions.

further investigation to build more robust systems. We would like to continue to investigate various issues such as: where to place the predictions, how many candidates to show, and how long the predictions should persist on the screen with/without the user's interaction. It is also necessary to perform longitudinal studies in more realistic situations that require complex planning and composition with a more diverse age range of users. Investigating the possibilities of *predictive handwriting* for other languages and applying the system to them will also be a promising research direction. Comparing support ratios of Japanese/English contents, and multimodal/unimodal situations will reveal more about the nature of *predictive handwriting*.

ACKNOWLEDGEMENTS

This research was supported by IPA Exploratory Software Project 2005, and 21st Century COE Information Science and Technology Strategic Core Project, Japan.

REFERENCES

1. Abowd, Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. *IBM Systems Journal, Special issue on Pervasive Computing*, *38*, 4, pp. 508-530, 1999.
2. Accot et al, More Than Dotting the I's - Foundations for Crossing-based Interfaces. *Proc. CHI'02*, pp.73-80, 2002.
3. Anderson et al., A Study of Digital Ink in Lecture Presentation. *Proc. CHI'04*, pp.567-574, 2004.
4. Apitz et al, CrossY: A Crossing-based Drawing Application. *Proc. UIST'04*, pp.3-12, 2004.
5. Clarkson et al., An Empirical Study of Typing Rates on mini-QWERTY Keyboards. *CHI'05 Extended Abstracts*, pp.1288-1291, 2005.
6. Darragh et al., The Reactive Keyboard: A Predictive typing aid. *IEEE Computer 23*, 11, pp.41-49, 1990.
7. Davis et al., NotePals: Lightweight Note Sharing by the Group, for the Group. *Proc. CHI'99*, pp. 338-345, 1999.
8. Denoue et al., Shared Freeform Input for Note Taking across Devices. *Proc. CHI'03*, pp.170-171, 2003.
9. Elrod et al., Liveboard: A Large Interactive Display Supporting Group meetings, Presentations, and Remote Collaboration. *Proc. CHI'92*, pp.599-607, 1992.
10. Feng, An Integrated Mulimedia Environment for Speech Recognition Using Handwriting and Written Gestures. *Proc. HICSS'03*, pp. 128b, 2003.
11. Fridland et al., Teaching with an Intelligent Electric Chalkboard. *ACM SIGMM Workshop on Effective Telepresence*, pp.16-23, 2004.
12. Fukushima et al., A Predictive Pen-Based Japanese Text Input Method and Its Evaluation. *Transactions of Information Processing Society of Japan 37*, 1, pp.23-30, 1996, in Japanese.
13. Getschow et al., A Systematic Approach to Design a Minimum Distance Alphabetical Keyboard. *Proc. RESNA'86*, pp.396-398, 1986.
14. Goto et al., Speech Completion: On-demand Completion Assis-tance Using Filled Pauses for Speech Input Interface. *Proc. ICSLP'02*, pp.1489-1492, 2002.
15. Goto et al., Speech Spotter: On-demand Speech Recognition in Human-Human Conversation on the Telephone or in Face-to-Face Situations. *Proc. ICSLP'04*, pp.1533-1536, 2004.
16. Hindus et al., Ubiquitous Audio: Capturing Spontaneous Collaboration. *Proc. CSCW'92*, pp.210-217, 1992.
17. Iwata et al., A Study on the Participation Method of Distant Learners into the IT-supported Lecture Using an Interactive Electric Whiteboard. *Transaction of Information Processing Society of Japan*, *2002*, 119, pp.33-40, 2002, in Japanese.
18. Kaiser, Multimodal New Vocabulary Recognition through Speech and Handwriting in a Whiteboard Scheduling Application. *Proc.IUI'05*, pp.51-58, 2005.
19. Kam et al., Livenotes: A System for Cooperative and Augmented Note-Taking in Lectures. *Proc. CHI'05*, pp.531-540, 2005.
20. Kawahara et al., Recent Progress of Open-source LVCSR Engine Julius and Japanese Model Repository. *Proc. CSLP'04*, pp.3069-3072, 2004.
21. Kristensson et al., SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. *Proc. UIST'04*, pp.43-52, 2004.
22. Lin et al., Chinese Character Entry on Mobile Phones: A longitudinal Investigation, *Interacting with Computers*, *17*, 2, p121-146, 2005.
23. Lyons et al., Augmenting Conversations Using Dual-Purpose Speech. *Proc. IST'04*, pp-237-246, 2004.
24. Masui, An Efficient Text Input Method for Pen-based Computers. *Proc. CHI'98*, pp.328-335, 1998.
25. Microsoft Mobile PC and Tablet PC Developer Center, http://msdn.microsoft.com/mobility/tabletpc/
26. MyFont, software by TechnoAdvance Co., Ltd., http://www.techno-advance.co.jp/product/myfont/
27. Ogata et al., Speech Repair: Quick Error Correction Just by Using Selection Operation for Speech Input Interface. *Proc. Eurospeech'05*, pp.133-136, 2005.
28. Oviatt et al., Individual Differences in Multimodal Integration Patterns: What Are They And Why Do They Exist?. *Proc. CHI'05*, pp.241-249, 2005.
29. Oviatt, Mutual Disambiguation of Recognition Errors in a Multimodal Architecture. *Proc. CHI'99*, pp.576-583, 1999.
30. Pedersen et al., Tivoli: an Electric Whiteboard for Informal Workgroup Meetings. *Proc. CHI'93*, pp.391-398, 1993.
31. PowerPoint, software by Microsoft Corporation, http://www.microsoft.com/office/powerpoint/prodinfo/
32. Schilit et al., Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations. *Proc. CHI'98*, pp.249-256, 1998.
33. Stifelman et al., The Audio Notebook. *Proc. CHI'01*, pp.182.189, 2001.
34. Vertanen, Efficient Computer Interfaces Using Continuous Gestures, Language Models, and Speech. *M.Phil Thesis*, University of Cambridge, 2004.
35. Wang et al., Chinese Input with Keyboard and Eye-tracking: An Anatomical Study. *Proc. CHI'01*, pp.349-356, 2001.
36. Wilcox et al., Dynomite: A Dynamically Organized Ink and Audio Notebook. *Proc. CHI'97*, pp.186-193, 1997.