Web-Based Applications Using Pen-Based Interfaces and Network-Based on-line Handwriting Recognition

Takeshi Sakurada, Mitsunori Yorifuji, Motoki Onuma, Masaki Nakagawa

Tokyo University of Agriculture & Technology
Naka-cho 2-24-16, Koganei-shi, Tokyo, 184-8588, Japan
takes@hands.ei.tuat.ac.jp

Abstract

This paper presents several applications and their user interfaces that employ pen-based interfaces and network-based on-line handwriting recognition. Pen-based interfaces are again attracting people’s attention since several platforms are now available such as Microsoft Tablet PC, Anoto pen and paper, E-pen and many more. We are interested in employing them for WWW applications. We have developed several pen-based applications such as handwritten memo application, questionnaire answering application, crossword puzzle application, etc. All the applications send handwriting to the handwriting recognition server when necessary to have it recognized by the server so that they do not need to employ handwriting recognition themselves. We discuss their merits and the UI design based on the client-server handwriting recognition architecture.

1 Introduction

Pen-based interfaces are attracting people’s attention again due to the wide spread of mobile PDAs (Personal Digital Assistant) with pen input, expanding sales of interactive electronic whiteboards, announcement of Anoto Pen from Anoto AB, shipping of Tablet PC of Microsoft Corp., etc.

Pen input has several benefits. It does not require large space. People can use it without training. Pen input on interactive whiteboards can easily attract the attention of an audience. People can express or annotate their ideas without being bothered by how to use it. Thinking is not interrupted by the actions for writing.

There are limitations and restrictions, however, on hardware and systems software (operating systems: OS) on which pen-based applications can be developed by employing a powerful handwriting recognition engine. Moreover, it seems almost impossible to develop consistent applications across the variety of environments employing the same quality or level of handwriting recognition without introducing new software architectures.

This paper presents several applications and their user interfaces (UI) that employ pen-based interfaces based on the network-based software architecture composed of an on-line handwriting recognition server and applications running on a variety of hardware and systems software. These applications can invoke handwriting recognition independent from the environment.

2 Problems of developing applications with handwriting recognition

We have been developing pen-based applications [M.Nakagawa(1999), H.Bandoht(2000),] using our own handwriting recognition engine [O.Velek(2002)]. In order to develop these applications,
however, programmers must know how to incorporate the recognition engine. Moreover, sufficient hardware resources of CPU power and memory space are necessary for installing our recognizer. Therefore, it is hard to port the latest recogniser into a small PDA without sacrificing part of the performance.

Another problem is concerned with the software platform used to develop and run pen-based applications. We have been employing Microsoft Windows OS. This is the most powerful platform, but poses a problem when transporting the applications to other operating systems.

Due to these problems, we have chosen to employ a client-server architecture based on a server-side handwriting recognition engine and client applications, and develop applications assuming Web environment as a new common platform.

In the following sections, we begin with Web-based applications, and then present the client-server scheme for handwriting recognition systems through networks. Applications can be developed without including a handwriting recognition engine and executed independent of the problem of systems environment and hardware resources.

3 Prototyping of Web-based applications

We made several pen-based client applications, some of which are Web-based and others are developed in Java. These applications can be executed independent of operation systems. In the subsequent sections, we will show the following applications; Freehand Notepad, Questionnaire Answering, Crossword Puzzle, and Mosaic Kanji Game.

3.1 Freehand Notepad

Fig.1 shows a screenshot of our Freehand Notepad application on which one can take notes freely in handwriting and have the notes be recognized when necessary. The free handwriting recognition requires character segmentation, character recognition, and context processing so that its demand on CPU power is very large. By making the recognition server solve these tasks, clients are freed of any CPU requirement.

3.2 Handwriting Questionnaire Answering

Fig.2 shows a screenshot of our Handwriting Questionnaire Answering system. A writer can tick one of several possible answers to a question with his electric pen, after which the tick is automatically recognized. In case of free text input, the user can write any text, which is then recognized by the handwriting recognition server. This application allows users to answer questionnaires like on a sheet of paper, on a Tablet PC or PDA. The answers are collected and summed up into a report automatically.

Fig.1 Freehand Notepad.

Fig.2 Handwriting Questionnaire Answering.
3.3 Crossword Puzzle

Fig. 3 shows a screenshot of our Crossword Puzzle Game. A player can write a letter in a blank square and the software recognizes and displays the printed character. This provides a far better interface than mouse and keyboard, since a player can write a letter by simply moving its hand and writing.

3.4 Mosaic Kanji Game

Fig. 4 shows a screenshot of our Mosaic Kanji Game. Although a player may know the rough shape of a kanji (Chinese character) and how to read it, it is often difficult to actually write the character. This game is for learning how to write Kanji correctly.

The player looks at the tessellated kanji on the left side and tries to write the kanji in the right box, guessing from the rough shape. Then, the application tells the player if his/her handwriting is the correct answer by asking the handwriting recognition engine. One's memory of writing is often vague, especially for complicated kanji patterns.

Asian children must be able to write thousands of kanji patterns, so they have to spend some time almost everyday to write and memorize Kanji patterns. This application helps them learn Kanji patterns with fun even on a small PDA.

4 Network-based On-Line Handwriting Recognition System

The above-mentioned applications using handwriting input do not have integrated handwriting recognition. Instead, they access a handwriting recognizer through a network. In the following subsections, we will describe our client-server architecture for handwriting recognition.

4.1 System Requirements

Handwriting recognition systems using networks have been developed to provide the most powerful handwriting recognition to even small and slow systems without sufficient CPU power or memory resources [T.Sakurada (2003)].

Current products with Japanese handwriting recognition on the market employ character writing frames to avoid the segmentation problem where a user writes each character into a separate writing frame to get it recognized. To not only recognize very casually written and deformed patterns, but also to recognize freely written text without writing frames, we have developed a highly reliable recognizer based on a combination of on-line and off-line recognition methods.
[O.Velek(2002)] as well as a frame-free, character-orientation-free and line-direction-free handwriting recognition system [M.Onuma(2003),T.Oki(2003)]. The users do not need to care about the input frames, character orientation, line direction or size of characters. They can naturally write and input characters like writing on a sheet of paper with a pen. However, this advanced handwriting recognition engine requires far larger CPU power and memory resources, so it is difficult to run it on a PDA while satisfying real-time constraints.

On the other hand, PHS and wireless LAN make the Internet accessible anytime, anywhere. Therefore, we employed a client-server scheme for handwriting recognition. The very powerful frame-free handwriting recognition is available from any system, requiring only Internet connection capability, so that even mobile terminals with little CPU power and small memory resources can apply sophisticated handwriting recognition by using the network infrastructures.

4.2 Outline of the Client-Server System

Fig.5 shows the overview of the system. The system consists of three parts: client agent, server agent, and recognition engine. The client agent accepts handwriting input and sends it to the server agent. The server agent invokes the recognition engine to get candidates for the recognition result of the inputted pattern and return them to the client agent. The client agent displays them to the user. The user chooses the intended answer from the candidates. If necessary, the user corrects the pattern, which is then resent to the server, and the input process starts again.

In the actual experiment with an 28.8kbps analog modem, we have verified that our system performs input-frame-free character string recognition and returns recognition result in real-time. Moreover, we have also observed that our client-server architecture processes handwriting patterns at much higher speed than many of the clients running their own client-side character recognition.

![Diagram](image)

Fig.5 The overview of the system.

4.3 Advantages of the architecture

When this client-server system is employed, it becomes unnecessary to prepare a recognition engine for every environment on the client side. Therefore, the recognition engine imposes no constraints on the client’s hardware and software, and the development of recognition engines for different environments becomes very cheap. Moreover, this architecture allows us to develop applications without understanding the details of a handwriting recognition engine.

Improvements or modifications of the handwriting recognizer require only changes on the server side. Once the changes are made, all clients can immediately and simultaneously benefit from these adjustments.
Different client environments can share one handwriting recognition engine, so that a writer can customize all environments simultaneously. For instance, when a person uses two or more PDAs and Tablet PCs, improvements of the recognizer are automatically distributed to all environments rather than requiring the writer to spend time and effort to update all of them independently. Storing all the handwritten patterns and their accepted or corrected recognition results on the server enables an even more powerful customization. In this case, the database of handwritten patterns is automatically expanded allowing continual training of the recognition engine.

5 Conclusion

This paper presented several applications and their user interfaces (UI) that employ pen-based interfaces and network-based on-line handwriting recognition. This architecture frees applications of incorporating a handwriting recognition engine and thus of hardware and software requirements imposed by the engine. We developed client applications assuming the Web environment as a common platform to ensure portability as high as possible. All applications shown here take advantage of the combination of pen interfaces and our client-server architecture for handwriting recognition.

Acknowledgements

This work is being supported by the program “The R&D support scheme for funding selected IT proposals” of the Ministry of Public Management, Home Affairs, Posts and Telecommunications.

References


