

A User Interface Using Fingerprint Recognition - Holding Commands and Data Objects on Fingers -

Atsushi Sugiura Yoshiyuki Koseki

C&C Media Research Laboratories, NEC Corporation
4-1-1 Miyazaki, Miyamae-ku, Kawasaki 216-8555, JAPAN
E-mail: sugiura@ccm.cl.nec.co.jp

ABSTRACT

This paper describes a new user interface, called a *fingerprint user interface* (FUI), which employs fingerprint recognition. While the unique feature patterns of fingerprints have mainly been used for personal identification, the FUI is unique in that it uses not only the differences among fingerprint patterns of different persons, but also the differences among the ten fingers of a single person. In the FUI, the system identifies the finger that has operated (touched) an input device through pattern matching of fingerprints and it performs the task assigned to the identified finger. Since users are able to specify different tasks by using different fingers, they feel as if commands and data objects were actually held on their fingers.

KEYWORDS: Fingerprint recognition, multimodal user interfaces, multi-computer user interfaces, input devices.

INTRODUCTION

Most electronic machines, such as computers, industrial control systems and appliances, are designed so that they can be operated by pushing buttons (keys) or by touching touch-sensitive screens with *fingers*. Although there are a lot of opportunities that fingers touch input devices, the interactions between the fingers and the input devices are very poor. On a button-based human machine interface, for example, a user's finger simply provides a trigger for executing commands assigned to the button.

This paper proposes a new user interface, called a *fingerprint user interface* (FUI), that enhances finger-machine interaction capability by using fingerprint recognition technique. The system identifies which finger is being used to manipulate (touch) an input device with fingerprint recognition, and it performs the task that has been assigned to that finger. This framework of a FUI enables a user to virtually hold commands in his/her own fingers. Since a user is able to indicate different

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST '98, San Francisco, CA

© 1998 ACM 0-58113-034-1/98/11... \$5.00

commands through the choice of the finger even in pressing a single button, the user can operate the system as if he/she could transfer the commands, attached to his/her fingers, to the system. Similarly, the fingers can virtually hold data objects. With the FUI, an object selected on an editor can be assigned to a specific finger that touches a "store in finger" button and it can be inserted into the editor only when the same finger touches a "get from finger" button. In this case, user's fingers can function as a data storage that exchanges data objects with the system.

In order for the FUI to be able to capture the fingerprint pattern when a finger has touched an input device, fingerprint scanners would naturally have to be included in the construction of such devices, e.g. on the button surfaces. At present, however, fingerprint scanners are often too large to be included in that ways within the user interfaces of such systems as hand-held computers and portable telephones. Nonetheless, in order to test the effectiveness of the FUI concept, we have developed a prototype FUI system consisting of a notebook computer and an optical-type fingerprint scanner.

In the following sections, we first consider the FUI in the light of related work, and next explain its concept, design, and usage. We also describe our prototype system, and finally discuss the usability and operability of the FUI.

RELATED WORK

Since fingerprints have unique feature patterns, they have been used for personal identification and authentication. Many security systems and entrance management systems are based on the fingerprint recognition technique [5] [6]. The FUI is the first attempts to apply the fingerprint recognition technique to the user interface field. It differs from the personal identification systems in that it uses not only the uniqueness of the fingerprint patterns of different persons but also that of the ten fingers of a single person.

From the viewpoint of user interfaces, a FUI can be categorized as a multimodal user interface [8] that enables a user to interact with a computer using such modalities as speech, gestures, facial expressions and eye direction [1] [2] [3] [11]. The FUI introduces a new

modality, *finger types*.

In the multimodal systems, the speech has been the most widely used modality. This is because it is the most natural means for human communications and it is able to convey various types of information. However, the speech-based interface is inappropriate for the use in public places, because it might disturb others and it might leak the contents of user's work. Although FUI has less capability for conveying information than the speech-based interface, it can be used in public without any trouble.

Pick-and-Drop [10] is a pen-based direct manipulation technique for multi-computer environments, which allows a user to virtually hold a data object in a pen that does not have storage capabilities. In Pick-and-Drop, an object, pointed by the pen on one display, is bound to the pen ID. When the pen next touches another display, that object is moved/copied to it via a network. That is to say, the user can manipulate the object as if it were stored in the pen.

Although the concept of a FUI for object manipulation is similar to that of Pick-and-Drop, it has several advantages. First, the FUI does not require special tools to manipulate objects, because users' fingers can be used in place of the ID-recognizable pen of Pick-and-Drop. Moreover, the FUI does not need the ID management, which is required in Pick-and-Drop. It becomes increasingly laborious to guarantee the uniqueness of pen-IDs, when the number of pens increases. The necessity of such an ID management task is eliminated with the FUI, because of the inherent uniqueness of fingerprints; this is an important advantage.

Another advantage of FUIs is that a user can possibly handle up to ten data objects simultaneously by using ten fingers. This can be done also in Pick-and-Drop by using ten different pens. However, using multiple pens is not a realistic approach. Furthermore, FUI enables a user to specify several different commands at a single input device, by assigning commands to fingers. Such a user interface cannot be built by a pen-based Pick-and-Drop

method.

FINGERPRINT USER INTERFACE

Basic Concept

A FUI is a user interface that employs fingerprint recognition. Using the FUI, a user can specify different tasks by using different fingers for operating an input device. Since all fingers of a single person have unique fingerprint patterns, the finger used for the operation can be identified through the matching of the fingerprint patterns.

Figure 1 shows the basic architecture of the FUI. The fingerprint image is captured when the finger touches the fingerprint scanner incorporated into the surface of an input device. If the captured image matches the one of the template fingerprint patterns registered in the *finger ID table*, the corresponding command or the data object is sent to the operation-target system.

This architecture for identifying a finger touching an input device leads to new types of interactions between fingers and input devices. One of the main characteristics of FUI is that users can manipulate commands and objects as if they were actually attached to the users' own fingers.

Suppose a push-button interface to start one application out of three. In a conventional user interface without fingerprint recognition, three different buttons are needed to specify three different applications (Figure 2a). With a FUI, however, a single button is sufficient to specify the application. It is determined depending on which finger pushes the button (Figure 2b). The user can indicate the applications as if he/she held them in his/her own fingers and sent them to the system.

Figure 3 shows another example of using a FUI. When the finger touches the fingerprint scanner, the texts assigned to each user's finger are inserted into a document in a text editor. This makes the user feel as if the data were being stored in the fingers and it were copied from the fingers to the editor. The user can use his/her own fingers just as a virtual data storage.

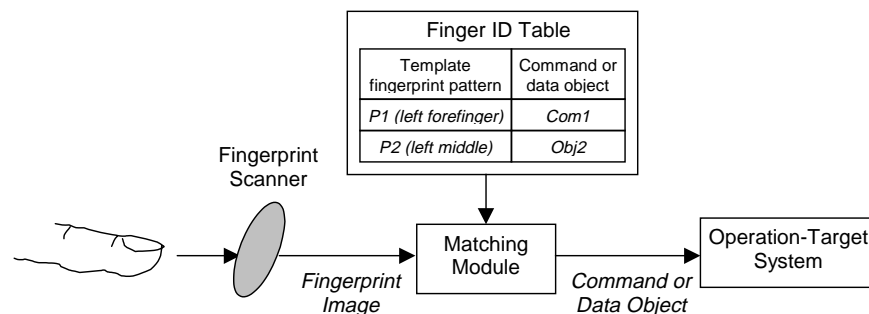


Figure 1: Architecture of fingerprint user interface.

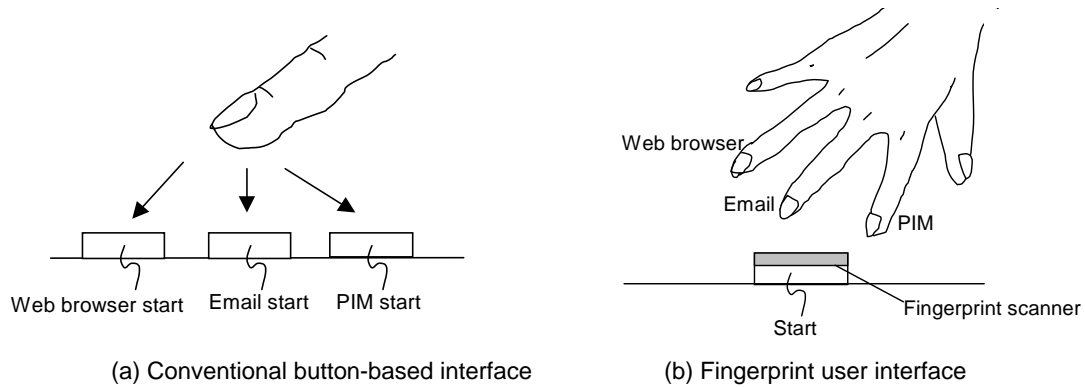


Figure 2: Comparison between two types of user interfaces.

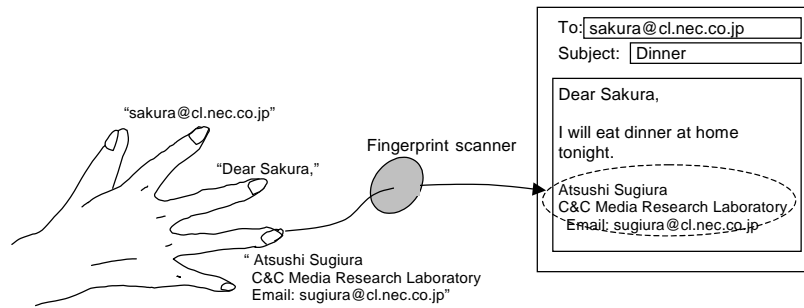


Figure 3: Using fingerprint user interface as virtual storage.

Designing a Fingerprint User Interface

As shown in Figure 1, the core components that compose a FUI are fingerprint scanner, the matching module for the fingerprint recognition, and the finger ID table. In addition to these, we incorporate two functions into the FUI to facilitate using it: visualizing commands/objects assigned to fingers and animating their virtual flow.

Incorporating Fingerprint Scanners: The most critical issue in the FUI design is how fingerprint scanners can be incorporated into input devices. Since the fingerprint pattern is captured when the finger touches the input device, it is required that the scanner should be put onto the surface of the device. So, the range of applications, to which FUIs can be applied, depends foremost on the size of the fingerprint scanners.

The most popular scanner to obtain a live-scan fingerprint image is an optical-type scanner, which consists of an assembly of an LED light source and a CCD placed on the other side of the glass plate. The CCD camera captures a fingerprint image illuminated by the LED light source. Since, however, the optical-type scanner needs some volume to ensure the light reflection, there is a certain limit for the size reduction. Actually, the current average thickness of the optical-type fingerprint scanner is 3-4 cm. This prevents the scanner from being incorporated with the push-button surface of such small machines as a portable telephone and a hand-held computer. It can, however, be used as a *touch-sensitive*

*button*¹ embedded in larger machine. Our current experimental system uses an optical-type scanner (details later).

Semi-conductor fingerprint scanners [7] [9] [12] have the possibility to solve the size problem. The scanner of Veridicom [13], based on sensing differential capacitance, is only 3 mm thick. This could be incorporated into a push-button surface. The reduction in the scanner size would expand the application domain of FUI.

However, these currently available scanners do not allow direct manipulations of objects in FUIs. To make the direct manipulations possible, a touch-sensitive screen must have a fingerprint scanning capability. If such a screen were developed, the user could pick up an object, pointed by a finger, directly from a screen to the finger. Also, the user could send a message (command) from a finger to the pointed object.

Fingerprint Recognition: The fingerprint, obtained by the scanner, is identified using pattern recognition techniques. Many fingerprint identification methods have been proposed for the purpose of personal identification, and two main methods are currently available: the minutiae method based on the matching of fingerprint feature patterns (ridge ending patterns and ridge

¹ In this paper, we call a button-like input device, operated by touching it without actually pushing it, a *touch-sensitive button*.

bifurcation patterns of minutiae) [6], and the image matching method based on the shading patterns of a fingerprint image [14].

These two methods can be used in FUIs, but there is a trade-off between accuracy and recognition time. Generally, the minutiae method is more accurate but needs more time for recognition than the image matching method. In the minutiae method, it is necessary to extract fingerprint features from the captured image before the actual matching. This feature extraction process makes the recognition time longer.

Although the slow response of the minutiae method might irritate users, we chose the minutiae method for our experimental system because of its high accuracy of identification. We feel that user operations would suffer more from the misidentifications of fingerprints that cause the executions of undesired commands, rather than from the slow response of the system.

Constructing a Finger ID Table: In order to use the FUI, it is necessary to create a finger ID table in advance. That is, a template fingerprint pattern for each finger has to be registered, and the registered patterns have to be bound to the corresponding commands or data objects.

The template patterns can be registered by having the user pressing each finger onto the fingerprint scanner. Since the registered patterns can be commonly used from different applications installed on the same machine, the user does not have to register the patterns repeatedly for all applications. The commands and data objects can be assigned to the fingerprint patterns for each application separately.

In some cases, the template pattern can be naturally acquired through a sequence of user operations, without such pre-registrations. This can be seen in a *finger memo* application described later.

Visualizing commands and objects: One problem of FUIs is that commands and objects bound to fingers are invisible. In a graphical user interface (GUI) environment, users can surely specify their desired commands because the command names are displayed on icons or menus. In the FUI, however, the user is required to memorize the commands assigned to each finger. Also, FUI has the problem of misidentifying fingerprints. If a fingerprint is falsely identified, an undesired command will be executed. Therefore, FUIs should provide a mechanism that allows the user to cancel an operation halfway in case the system is about to execute unintended tasks.

In our design of FUIs, the command or the object assigned to the users' fingers is displayed on a screen so that the user can confirm it (Figure 4). In the configuration that a fingerprint scanner is incorporated into a push button, the command would be displayed when the finger touches the button surface. If the user-intended command is displayed, the user pushes the

button to actually execute the command.

When using a fingerprint scanner as a touch-sensitive button, the system intentionally employs a certain time difference between the finger first touching the scanner and the actual execution of the command, in order to give users the opportunity to confirm the command. Namely, it is executed after the user keeps touching the scanner for more than a certain fixed time period. If the undesired command is displayed, the user simply takes the finger off the scanner so that the command is not executed. Otherwise, the user keeps touching the scanner until the command has actually been executed.

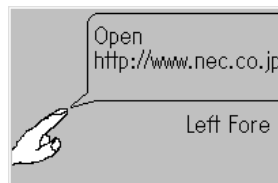
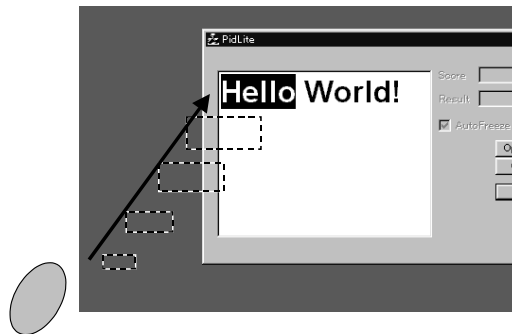


Figure 4: Dialog for displaying commands and objects held on fingers.

Animating virtual flow of commands and objects: When a task assigned to a finger is actually executed, a FUI animates the virtual flow of the command or the object from the finger (the nearest position on a screen to a fingerprint scanner) to an application, as shown in Figure 5. This animation is helpful for users to understand the concept of the FUI.



Fingerprint scanner

Figure 5: Animation of the virtual flow of an object.

Usage of Fingerprint User Interfaces

Although the concept of FUI is simple, it is useful in many situations. The usage of a FUI can be divided into two ways: assigning commands to fingers (*finger command*) and virtually storing data objects in fingers (*finger storage*). Table 1 and Table 2 show examples of the finger command applications and those of finger storage applications, respectively.

Usage as Finger Commands:

- **Modeless operations.**
The simplest usage of FUI is to support modeless operations. Many user interfaces employ mode conversion method to make it possible to specify many commands with fewer buttons or keys. In a GUI environment, such as Windows 95, object selection modes are converted by using key pressing. While a mode for single object selection is indicated by a simple click of a mouse button without key press, multi-selection and regional selection modes are indicated by pressing a **Shift** key and a **Ctrl** key, respectively. In a FUI, however, such mode conversions are unnecessary. If a fingerprint scanner is incorporated into the mouse button surface, the modes can be specified by finger types that click the mouse button (e.g. the regional selection is specified by a mouse click with the middle finger). This enables a single hand operation. It is not needed to use both hands for pressing the key and clicking the mouse button at the same time.
- **Operations without looking at an operation panel.**
A FUI also helps a user to manipulate input devices without looking at them. Consider the case of operating a car stereo while driving a car and looking directly ahead. In this case, it is difficult to find the target button without actually looking at the operation panel, even though the rough position of the stereo can be found by groping. In FUIs, if only the user memorizes the commands assigned to each finger, the user can operate the stereo by roughly finding its position and then simply touching it with different fingers. Moreover, the FUI is also useful for operating machines in darkness and manipulating them in a bag. It might also be useful for constructing user interfaces for the disabled.
- **Concealing commands from other persons.**
Another interesting characteristic of the FUI is to enable the user to conceal the command the user specified from other persons. Suppose the case, for example, where an exhibitor gives demonstrations to many different visitors at an exhibition. In exhibitions, it is common to give different demonstrations according to the titles and the departments of visitors. The FUI allows the exhibitor to choose different demonstrations simply by pushing a button with different fingers, not pushing obviously different buttons in front of the visitors. This prevents visitors from knowing that the exhibitor is discriminating. In another case, at the check out counters of convenience stores, cashiers have recently been inputting customers' sex and age judging from their appearance in order to obtain marketing data. With the FUI, these inputs can be hidden from the customers.
- **Minimizing the number of buttons.**

A FUI makes it possible to specify multiple commands by using only a single button. In other words, the FUI can minimize the number of buttons necessary in operations. This characteristic of FUI would be useful for the user interfaces of small machines that do not have enough spaces to place many buttons, such as pagers, portable phones and palm-top computers.

Usage as Finger Storage: In FUIs, fingers can be used as virtual data storage. For example, the fingers can be bookmarks for Web pages by binding uniform resource locators (URL) to each finger, that is, opening "http://www.xxx.com" by the forefinger, opening "http://www.yyy.com" by the middle finger, etc. Likewise, they can be a telephone directory and an address book. Similarly, attaching texts that are frequently used for creating documents to fingers (Figure 3) is effective in situations where text is difficult to input, for example, on pen-based user interfaces. Furthermore, fingers can be used as cut buffers that allow a user to keep multiple objects simultaneously.

The most interesting usage of a FUI is when a finger ID table, managing fingerprint patterns and data objects, can commonly be accessed from multiple computers connected to the network. This configuration enables a Pick-and-Drop-like operation on the FUI, called a *finger memo*.

Figure 6 shows the actual and virtual flows of data in the finger memo. When user's finger first touches the fingerprint scanner of one computer, the indicated object (the object selected on the display) is bound to the captured fingerprint and it is registered in the finger ID table. When the finger next touches another fingerprint scanner, the finger ID is identified through pattern matching of the fingerprints and the object bound to the identified finger is copied via the network. For the user, the object appears to be copied via the user's own finger. Note that, in the finger memo, it is unnecessary to register template fingerprint patterns and corresponding data objects to a finger ID table in advance.

The uniqueness in fingerprint patterns among different persons expands the application area of FUIs to kiosk terminals used by unspecified persons. Suppose that a finger ID table can be accessed from any kinds of electronic machines, such as kiosk terminals, public phones, and personal computers, as shown in Figure 7. In such environments, it is possible to copy a phone number in Email received at a home desktop computer to the finger, go out with holding the number on a finger and call the number using a public phone. On the contrary, if the user finds useful information on a kiosk terminal, such as an information terminal installed in a train station and a book retrieval system at a library, he/she can bring it back to a home desktop computer or a PDA. With the finger memo, pens and memos are no longer necessary to be carried.

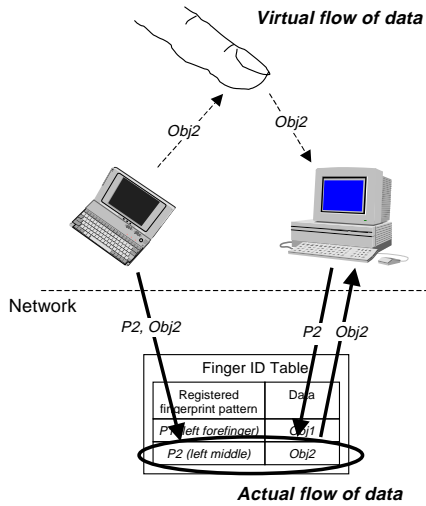


Figure 6: Finger memo.

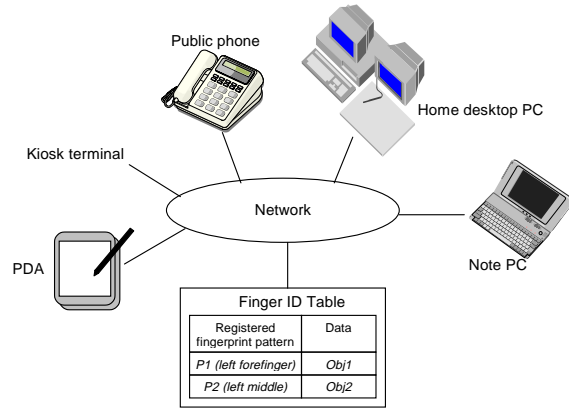


Figure 7: Use of fingerprint user interface on network environment.

Table 1: Usage of fingerprint user interface as finger command

| | |
|--|---|
| Modeless operations | <ul style="list-style-type: none"> - Object selection: mouse button click (single selection), Ctrl + click (multiple selection), Shift click (regional selection) - Object move: Shift•{ drag (move), Ctrl•{ drag (copy) - Drawing tools: changing figure types by finger types - Operating a CD player by finger type (play, stop, fast-forwarding, rewinding) |
| Operations without looking at operation panels | <ul style="list-style-type: none"> - Operations while driving a car - Operations in darkness - Operations of machines in a bag or a pocket - Operations while looking through a telescope or a microscope - User interface for the disabled |
| Concealing commands from others | <ul style="list-style-type: none"> - Selecting demonstration formats at an exhibition - Inputting marketing data at a check-out counter - Setting off alarms: opening a cash register by pressing a button with the forefinger, informing the police by pressing the button with the middle finger |
| Minimizing buttons | <ul style="list-style-type: none"> - Minimizing the number of buttons of portable computers - Unification of remote controllers of TV and Video |

Table 2: Usage of fingerprint user interface as finger storage

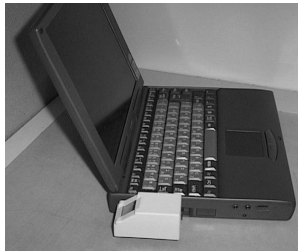
| | |
|-----------------------------------|--|
| Usage on a standalone environment | <ul style="list-style-type: none"> - Bookmarks for Web pages - Phone directory - Keeping frequently used strings for document creation - Multiple cut buffers |
| Usage on a network environment | <ul style="list-style-type: none"> - Finger memo (Pick-and-Drop without pens) - Calling the phone number attached to a finger on a public phone - Bringing back information from a public terminal and saving it on a home desktop computer |

EXPRIMENTAL SYSTEM

Implementation Details

We developed an experimental system to verify the usefulness of the FUI. In this system, we currently use an optical-type fingerprint scanner of NEC Corp. that can be inserted into a PCMCIA slot (Figure8). The size of the scanner excluding the card is 3.3 cm wide, 6.2 cm deep,

and 3.1 cm high. Because of this size, only one scanner can be inserted into the slot. Also, since the position of scanner is fixed in the PCMCIA slot, it cannot be used as a push button, but only as a touch-sensitive button. When a user's finger touches the scanner, the system is notified of the image-capturing event. The interface between the scanner and the applications is provided by an ActiveX



(a)

Figure 8: Experimental system and fingerprint scanner.



control, and the applications described in the next subsection were developed with C++ on Windows 95.

To identify the fingerprint, the system employs the minutiae method [6]. A matching score is calculated by comparing the feature pattern of the captured fingerprint with that of the template pattern. If the score is above a threshold value, the fingerprint is identified.

The computer used is a B5-size notebook with 133 MHz Pentium processor. With this 133 MHz processor, about 1.6 seconds are required for extracting features from the captured fingerprint image, and about 0.06 seconds for the matching of one fingerprint. So, if there are n registered template fingerprints, it takes $1.6+0.06*(n+1)/2$ seconds on average to identify the finger (from scan to matching).

Example Applications

Due to the limitations on the experimental system, only limited types of applications can currently be built. We made three example applications: an operation of CD player, a finger bookmark, and a finger memo.

Operating a CD Player: The first application is an example of a finger command application to operate a music CD player according to which finger touches the fingerprint scanner. As a CD player, we used the one provided by Windows 95 (installed as `cdplayer.exe`). The play, stop, fast-forwarding and rewinding are specified by using the left index finger, the middle one, the ring one, and the little one, respectively. Since these commands are originally bound to finger types in this application, the user simply has to register template fingerprint patterns to use this application.

Finger Bookmark for Web Browsing: A finger bookmark is a URL repository built on user's fingers. When the finger touches the fingerprint scanner, the Web page assigned to the finger is displayed on a Web browser. With the finger bookmark, the user is not required to open the browser's bookmark and search it for the target URL.

The finger bookmark not only has the feature of a finger storage holding the URLs in each finger, but also has that of a finger command operating as shortcuts for opening Web pages.

Finger Memo: A finger memo is a finger storage

application that allows the user to transfer a data object among multiple computers by virtually picking it up with his/her finger. We implemented it using two notebook computers, each of which employs a fingerprint scanner. These two computers refer to a common finger ID table placed on the network.

With the FUI technique, the user can transfer the object between two computers without using a pen that is necessary in the original Pick-and-Drop. However, our finger memo does not offer the capability of direct manipulation of objects. In the Pick-and-Drop, the user can pick the target object directly from a touch-sensitive screen. To make direct manipulation possible in FUIs, we would need a screen with the capability of scanning the fingerprint when the finger touches it. In our current implementation, the objects to be picked up are indicated by using a mouse. The user first selects the object by the mouse and next touches the fingerprint scanner to virtually pick the selected object. On another computer, the user indicates the position where the object is dropped using the mouse, and touches the scanner to actually drop it.

DISCUSSION

Usability as Finger Storage

Even in the current configuration of the experimental system, users feel as if their fingers could hold data objects by using the finger memo and the finger bookmark. Although the finger memo does not have the capability for direct manipulation, its usefulness is obvious because it does not require a special tool, such as an ID-recognizable pen of Pick-and-Drop.

As we mentioned, the finger storage is feasible even in kiosk terminals in public spaces, used by anyone, without the complex ID management of fingerprints. If all electronic systems in a city, such as public phones and electronic bulletin boards, would be connected to the network, users could take objects anywhere with their fingers.

Usability as Finger Command

The finger command user interfaces built on the experimental system do not have distinct advantages over the usual GUI environments, such as the Windows 95 desktop where a mouse and a keyboard are available.

The usefulness of finger bookmark is almost the same as that of the URL shortcuts created on the Windows

desktop. In the case of the CD player application, it is better to click “▶ (play)”, “■ (stop)”, “▶▶ (fast-forwarding)” and “◀◀ (rewinding)” buttons by the mouse than to use different fingers to operate the CD player. In general, assigning a single function to a single button is more natural than assigning multiple functions to a single button.

The finger commands are expected to be useful in somewhat special situations: operations on a small-sized machine where a sufficient number of software icons and hardware buttons cannot be placed, and operations of a machine in a bag where users cannot look at the operation panel. Suppose a case, for example, where a user browses Web pages on a small-screen computer while standing in a moving train. Since there is not enough space to place URL shortcuts on the screen, the user is required to start up a browser, open a bookmark, and search the list for the target Web page. Such Web browsing tasks in the train are heavy burden for the user. The finger bookmark on the FUI would enable users to specify the target Web page by a single action, just pushing the “Open URL” button with appropriate fingers.

Also, suppose a case where a user manipulates a headphone stereo in a bag, like Sony Walkman, groping its operation panel. In this case, the fingerprint user interface of our CD player application would be useful. We conducted an informal experiment regarding the operations of machines in a bag without watching the operation panel. We observed that it was easier for subjects to push a single button by using different fingers than to grope the target button out of multiple buttons on the panel.

A double click of a mouse button is another way to specify multiple commands using a single button. However, only two commands can be distinguished by using single and double clicks. Although the triple click allows users to deal with more than two commands, it is difficult to click the button many times at short intervals. With the FUI, the user can specify up to ten commands by a single click (or touch) of an input device.

Operability

Through the verification on the experimental system, it became clear that the FUI is inappropriate for tasks where a fast input speed is required. To press a single button by one of multiple fingers, the user must move the palm position. Since the palm position relative to the input devices cannot be kept constant, the FUI is not expected to increase the input speed.

Furthermore, FUIs are unsuitable for the single hand operations of palmtop machines. When the relative position between the palm and the input devices is fixed, it is difficult to move only the fingers right and left. In the FUIs, it is necessary to operate a machine using both hands (one hand for holding the machine and the other for manipulating the input device), or the user has to set

the machine down and operate it using a single hand.

In the current FUI implemented on the experimental system, users have to take care when pressing fingers on the fingerprint scanner. In order that the system can acquire enough number of features of fingerprint patterns (ridge endings and ridge bifurcations) to identify the fingerprint, a large area of the finger has to be placed on the scanner. In the practical use, however, only a fragment of the fingerprint image might be captured if the user carelessly touches the scanner. For practical applications, it would be necessary that a finger type be correctly identified only from a fragment of the fingerprint image.

Security

A critical issue in using FUIs is to ensure the security against illegal use of stolen fingerprint data. Especially, in the finger memo built on an open network, there is dangerousness that the fingerprint data is stolen. However, we consider that users do not seriously suffer from the steals of the fingerprints for the following reasons.

When using fingerprint data on the network, it should be encrypted. Although there are still some fears that the encryption is broken, the fingerprint data cannot be mapped to the person’s name in the finger memo application.

Moreover, even if the mapping is done unfortunately, we speculate that the stolen fingerprint data cannot be used from the network for the authentication of the mission-critical systems, such as entrance management systems, ATMs of the banks, credit card systems. This is because these systems are built on the closed private network until completely secure encryption methods are developed. In other words, the systems that users seriously suffer from the steals of fingerprints are not used on public networks².

Also, if replica fingerprint images, generated by decoding the stolen fingerprint data, try to be used on the fingerprint scanners, the system can reject such illegal use by using techniques that discriminate between living fingers and replicas [4].

However, we should discuss this issue carefully because users cannot change their own fingerprint patterns just as changing passwords.

CONCLUSION AND FUTURE WORK

This paper proposes a new user interface technique using fingerprint recognition. The proposed fingerprint user interface allows users to manipulate objects and commands as if they were actually attached to their

² In the finger memo, there are also dangers that data is illegally picked up and dropped using the stolen fingerprint pattern. However, the dangerousness of using the encrypted fingerprint data is the same as that of a password-based security method.

fingers. Although the concept of the fingerprint user interface is simple, it can be used in many situations.

One of our future works is to apply the FUI technique to user interfaces of various systems. We have evaluated only a few applications on the experimental systems using the optical-type fingerprint scanner. We should be able to develop more applications by incorporating semiconductor scanners into small input devices and evaluate the effectiveness of FUI in various environments.

To make a FUI practicable, it is necessary to reduce the time needed for the fingerprint recognition. In our current system, it takes at least 1.7 seconds to identify a single fingerprint. This long response time irritates users, especially when they have to press the scanner repeatedly, for example, for operating the CD player.

As mentioned above, the matching algorithm for fingerprint patterns needs to be enhanced in order to improve the operability of a FUI. So, even with only a small number of features of the fingerprint pattern are given, the fingerprint should be correctly identified.

Although this paper focused on a user interface technique that uses the differences of a single person's fingers, the use of fingerprints for personal identification are also relevant. Fingerprints can be a useful means for personalization when multiple persons use a single machine. We will investigate this issue in our future research.

ACKNOWLEDGEMENTS

We would like to thank Kaoru Uchida and Huang Lei for supporting the system development. We also would like to thank Tomonari Kamba, Yuichi Koike, Hidekazu Sakagami, Hisashi Shimamura and Motohiko Sakaguchi for helpful discussions. We also express our appreciation to Satoshi Goto and Shiro Sakata of NEC Corporation for giving us the opportunity to pursue this research.

REFERENCES

- [1] Bolt, R.A., "Put-That-There: Voice and Gesture at the Graphics Interface", *Computer Graphics*, Vol. 14, No. 3, pp. 262-270, 1980.
- [2] Dohi H. and Ishizuka M. "Visual Software Agent: A Realistic Face-to-Face Style Interface connected with WWW/Netscape", *Proceedings of IJCAI'97 Workshop on Intelligent Multimodal systems*, pp.17-22, 1997.
- [3] Frey L. A., Preston White Jr. K., and Hutchinson T. E., "Eye-Gaze Word Processing", *IEEE Transactions on systems, Man, and Cybernetics*, Vol. 20, No. 4, 1990.
- [4] Fukuzumi, S. "Discrimination of Living Body Fingers by Using EMG and Finger Pulse Volume," *Proceedings of HCI International*, pp.575-578, 1997.
- [5] Hartman. M., "Compact fingerprint scanner techniques", *Proceedings of Biometric Consortium 8th Meeting*, 1996.
- [6] Hoshino Y. , Asai K., Kato Y., and Kiji K., "Automatic Reading and Matching for Single Fingerprint Identification", *Proceedings of The 65th International Association for Identification Conference*, 1980.
- [7] Klett J., "Thermal Imaging Fingerprint Technology", *Proceedings of Biometric Consortium 9th Meeting*, 1997.
- [8] Maybury M.T., "Intelligent Multimedia Interfaces", *The AAAI-Press/MIT-Press*, 1993.
- [9] McCalley K., Setlak D., Wilson S., and Schmitt J., "A Direct Fingerprint Reader", *Proceedings of CardTech/SecureTech*, Vol. 1: Technology, pp.271-279, 1996.
- [10] Rekimoto J., "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments", *Proceedings of UIST'97*, pp.31-39, 1997.
- [11] Sadek, M.D, Ferrieux A., Cozannet A., Bretier P., Panaget F., and Simonin J., "Effective Human-Computer Cooperative Spoken Dialogue", *International Conference on Spoken Language Processing*, 1996.
- [12] Schneider J., "Improved Image Quality of Live Scan Fingerprint Scanners Using Acoustic Backscatter Measurements", *Proceedings of Biometric Consortium 8th Meeting*, 1996.
- [13] Veridicom: <http://www.veridicom.com/>
- [14] Watson C. I., Candela G. T. and Grother P. J., "Comparison of FFT Fingerprint Filtering Methods for Neural Network Classification", *NISTIR 5493*, National Institute of Standards and Technology, 1994.