NaviPoint: An Input Device for Mobile Information Browsing

Kiyokuni KAWACHIYA and Hiroshi ISHIKAWA
IBM Research, Tokyo Research Laboratory
1623-14, Shimotsuruma,
Yamato, Kanagawa 242-8502, Japan
+81 462 73 4938
kawatiya@trl.ibm.co.jp

ABSTRACT
A mobile computing environment imposes various restrictions on users. For example, most mobile devices have a limited screen size, and it may be difficult to watch the screen closely. While the user is walking or standing in a bus or train, he or she may have only one hand free to manipulate the device. Therefore, some new operation method must be developed for comfortable information browsing in the mobile environment. In this paper, several existing methods are first introduced and compared from the viewpoint of their applicability in a mobile environment. A new input device for such an environment, named "NaviPoint," is then introduced. NaviPoint is a specialized device for mobile information browsing. By using this device, a user can perform three types of input — "analog input," "digital input," and "click input" — with just one finger. After an explanation of the conceptual structure and a qualitative analysis of NaviPoint, the structure of a prototype is described. Experiments using the prototype show that information browsing is possible with an overhead of less than 50% on the usual "mouse and scroll bar" method.

Keywords
Input device, information browsing, user interface, PDAs, hand-held devices, mobile computing

INTRODUCTION
With the evolution of compactization technology and the consolidation of the wireless communication infrastructure, it has become possible to access information through a network even while one is in transit from one place to another. In such an environment, it is considered that simple "information browsing" with no input of characters will become one of the major uses of mobile information devices. One typical example is access to hypermedia information such as the WWW. However, a mobile computing environment imposes various restrictions on users, and some new method of operation must be developed for comfortable information browsing. For example, a mouse cannot be used without a desk, and a pen, while of course indispensable for inputting characters, requires both hands even for simple browsing.

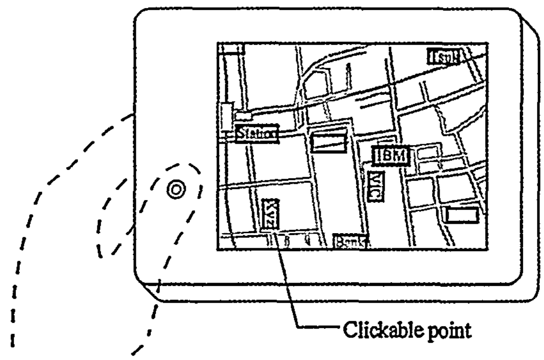

Figure 1. Example of mobile information browsing

This paper reports a new input device named "NaviPoint" that solves these problems and allows mobile information browsing with just one finger. The next section first introduces several existing operation methods for information browsing and describes the problems involved in applying them to a mobile environment. A new input device called NaviPoint that solves the problems is then introduced, and its basic mechanism is described. Next, the structure of a prototype is explained and its performance is evaluated. After a discussion of related work, the last section offers some conclusions and outlines future work.

INFORMATION BROWSING IN A MOBILE ENVIRONMENT
"Mobile information browsing" is becoming a major use of small mobile information devices such as PDAs [5]. One of the most general types of accessed information is "hypermedia information," which contains "clickable (selectable) points" that link it to other information. Figure 1 shows an example of such hypermedia browsing in a mobile environment. Part of a large map containing several clickable points is displayed. When one of these points is clicked, corresponding new information is displayed.

In recent years, as a result of the spread of the WWW, hypermedia browsing has become very popular in a desktop environment. However in a mobile environment, there are several restrictions. For example, most mobile devices have a limited screen size, and it may be difficult to watch the screen closely. The user may have only one hand free to manipulate the device. Comfortable information browsing requires some new operation method that includes an input device.[1]

Existing Operation Methods

Two fundamental operations for information browsing (hypermedia browsing) are "scrolling" and "pointing" [21]. The scrolling operation is necessary to browse information that lies outside the screen, and the pointing operation is necessary to select a clickable point on the screen. In most information browsing, the following two operations are performed repeatedly:

1. Browse the document by using the "scrolling" operation.
2. Select a clickable point by using the "pointing" operation.

Menu-based operations on PDAs can also be classified into the same category as scrolling and pointing if they require no input operation.

For these two basic operations, the following list summarizes several widely used existing methods.[2]

Method A. Mouse + scroll bars
: This is a standard operation method in the desktop environment. An arrow-shaped pointer (mouse pointer) is displayed on the screen. A scrolling operation is performed by dragging the slider of a scroll bar at the side of the screen (or window). A pointing operation is performed by moving the pointer to a clickable point and clicking.

Method A'. Trackball + scroll bars
: This operation method is generally used on notebook PCs. The basic scrolling and pointing methods are the same as in method A, except that a trackball (with an attached switch) is used instead of a mouse as an input device. There is another variation in which an analog input stick at the center of a keyboard is used instead of a trackball.

Method B. Pen + sheet-picking
: This operation method is used with most PDAs. A scrolling operation is performed by "picking" the document itself with a pen and dragging. A pointing operation is performed by directly pointing to (tapping) a clickable point in the screen.

Method C. Cursor keys + highlighting
: This operation method is mainly used with smaller mobile devices that are not equipped with a pen. One of the clickable points in the screen is "highlighted." A pointing operation is performed by moving the highlight to the target item by using up/down/left/right keys and pushing an execute key. Scrolling is performed implicitly according to this highlight movement, or by using a specially provided operation mode or keys.

Table 1. Comparison of existing operation methods

Reqs. / Methods	1. Arbitrary scrolling	2. Quick pointing	3. One-handed operation
A. Mouse + scroll bars	*Fair* (no angled scrolling)	Poor (close watching needed)	Poor (desk needed)
A'.Trackball + scroll bars	*Fair* (no angled scrolling)	Poor (close watching needed)	*Fair* (separate switch)
B. Pen + sheet-picking	*Fair* (1 screen in 1 op.)	**Good** (can be done directly)	Poor (two hands needed)
C. Cursor keys + highlighting	Poor (implicit scrolling)	**Good** (no slipping)	*Fair* (multiple keys)

The next subsection compares and examines these operation methods from the viewpoint of their applicability in a mobile environment.

Applicability to a Mobile Environment

Compared with a desktop environment, a mobile environment imposes various restrictions on user operations. Therefore, a specially designed operation method is necessary for comfortable information browsing. The following requirements should be satisfied:

Requirement 1. Arbitrary scrolling
: The scrolling operation is particularly important in a mobile environment, because the screen size is limited. For efficient operation, it should be possible to scroll in an arbitrary direction at an arbitrary speed.

Requirement 2. Quick pointing
: In a mobile environment, it may be difficult to watch the screen closely or continuously. Therefore, the pointing operation should move directly to a clickable point and select it.

Requirement 3. One-handed operation
: It should be possible to browse information even while walking or standing in a bus or train. It should therefore be possible to perform the above two types of operation with one hand.

Now let us evaluate the four operation methods described in the previous subsection on the basis of these requirements.

In methods A (mouse + scroll bars) and A' (trackball + scroll bars), the pointer itself can be moved in any direction. However, scrolling at an angle is impossible because these methods use scroll bars. In methods that display a pointer on the screen, continuous feedback from the screen is needed to move the pointer to a target item. Therefore, the pointing operation becomes very hard in an environment where the screen cannot be watched closely. First of all, an ordinary mouse cannot be used without a surface to press on, and is not suitable as an input device in a mobile environment. A trackball can be used in a mobile environment, but one-handed operation is somewhat difficult because it requires a separate switch.

[1] Of course, there are also various system-level problems such as how to utilize low network bandwidth or small memory capacity efficiently [12, 13], but these are beyond the scope of this paper.

[2] The four listed operation methods are typical examples. In addition, there are some combined methods and completely different methods.

In method B (pen + sheet-picking), the sheet-pick scrolling method allows scrolling in any direction. However, the length of one scrolling operation is limited by the screen size. The method allows the user to point to an item on the screen directly, but it always require two hands for operation in a mobile environment.

In method C (cursor keys + highlighting), comfortable scrolling is difficult, because it is performed implicitly. Even if some specialized operation mode is provided, it is still limited to at most eight directions and a fixed speed. On the other hand, pointing can be performed quickly and accurately because one clickable point is always highlighted. However, one-handed operation is difficult without special care, because the method requires multiple keys.

As we have seen, existing operation methods do not completely satisfy the three requirements for mobile information browsing. Table 1 summarizes the results of our analysis.

NAVIPOINT: A NEW INPUT DEVICE

Taking account of these requirements and problems, we have developed a new input device that allows comfortable one-handed mobile information browsing [6, 7, 8]. It is named "NaviPoint," because information navigation is facilitated by this pointing device.[3]

Basic Structure

NaviPoint is based on the "TrackPoint" analog input device [16] fitted to the center of the keyboard in IBM ThinkPads. A TrackPoint device can be used as a substitute for a mouse, and consists of a stick for two-dimensional analog input and several switches that correspond to mouse buttons. The stick senses horizontal forces but does not move physically.

As mentioned in the previous section, it is better for one-handed operation if the stick and switches are integrated. However, simply adding a switch at the bottom of the stick often causes unintentional pointer movement upon clicking. In NaviPoint, a physical "insensitive zone" is provided around the stick, allowing *click input* operation without any sensing of unintentional horizontal forces.

Figure 2 shows the conceptual structure of NaviPoint. NaviPoint consists of a stick with a micro-switch and a ring-shaped two-dimensional stress sensor around the stick (the left part of Figure 2). The stick can be depressed by applying a certain pressure, and can also be tilted in an arbitrary direction inside the ring. When the stick is tilted and presses against the ring, the direction and magnitude of the pressure are detected by the stress sensor. The hardware of this sensor is similar to that of the TrackPoint.

The right part of Figure 2 shows a state in which the stick is tilted. The ring-shaped part does not move physically. Therefore, the user knows when the stick touches the ring from the physical feedback of "contact feeling." At the

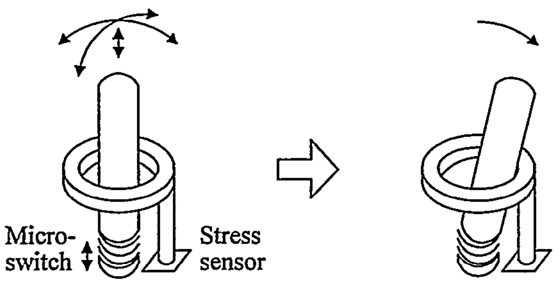

Figure 2. Conceptual structure of NaviPoint

Table 2. Qualitative analysis of NaviPoint

Methods \ Reqs.	1. Arbitrary scrolling	2. Quick pointing	3. One-handed operation
D. NaviPoint	**Good** (by analog input)	**Good** (by digital & click input)	**Very Good** (just one finger)

same time, the pressure on the ring is sensed. If the user returns the stick right away, a *digital input* operation is executed in that direction, like when a cursor key is pushed. But unlike with cursor keys, NaviPoint's digital input can specify an arbitrary direction. The software classifies the input into several directions (4 directions, 8 directions, etc.) according to the environment.

On the other hand, when the stick is tilted and pushes the ring continuously or firmly, an *analog input* operation corresponding to the force is performed, as in the original TrackPoint.

In a word, by using NaviPoint a user can perform three types of input — "analog input," "digital input," and "click input" — with just one finger. Because of the physical feedback from the micro-switch or the feeling of contact with the ring, a user can perform these input operations accurately without looking at the device or screen.

Information Browsing

To browse information with NaviPoint, use

- "Analog inputs" for scrolling,
- "Digital inputs" for changing the highlighted item, and
- "Click inputs" for selecting the highlighted item.

The basic method is similar to method C (cursor keys + highlighting) in the previous section, and one of the clickable points in the screen is highlighted. When the highlighted item goes off the screen as a result of a scrolling operation, the highlight is automatically moved to a clickable point on the screen. A typical information browsing procedure using NaviPoint is as follows:

1. Tilt the stick continuously (or firmly) to scroll the document.
2. Tilt and return the stick several times to move the highlight to the target item.
3. Depress the stick to select the highlighted item.

[3] The name has been changed from "ScrollPoint," which was used in [7, 8].

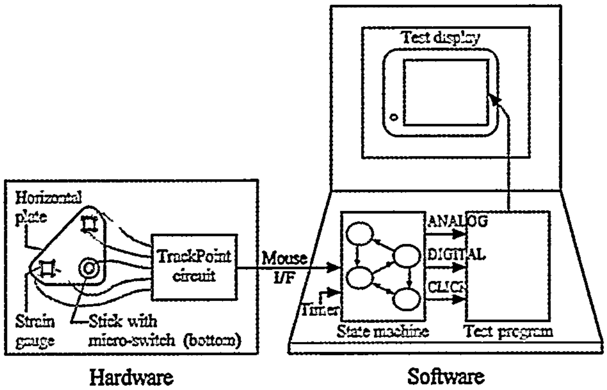

Figure 3. Structure of NaviPoint prototype

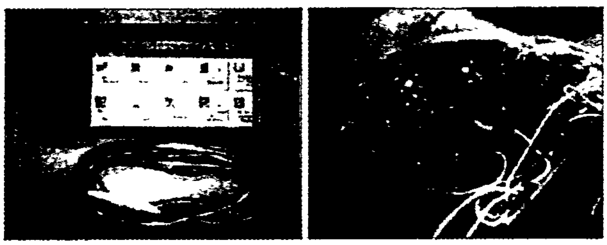

Figure 4. Prototype hardware (external and internal views)

Table 2 shows a qualitative analysis of NaviPoint based on the three requirements described in the previous section. The results show that NaviPoint is suitable for mobile information browsing. Especially as regards requirement 3, it has a big advantage in that it can be controlled *with just one finger.*

With isometric joystick-type input devices like TrackPoint, it is said, fine control is difficult and it takes time to point to a small target [11]. With NaviPoint, however, this is not a problem, because analog input is used only for rough scrolling of a document, and digital inputs are used to select a final target precisely.

THE NAVIPOINT PROTOTYPE

To show the efficiency of NaviPoint as an input device in a mobile environment, we made a prototype using TrackPoint parts. Figure 3 shows the structure of the prototype. The prototype consists of a hardware part (on the left), which implements the basic structure shown in Figure 2, and a software part (on the right), which analyzes the bare input from the hardware and categorizes it as "analog," "digital," or "click" input.

Hardware

The first prototype was developed to verify the feasibility of the NaviPoint concept. Therefore, the hardware was developed as a stand-alone device and is not compactized or built into an actual mobile device. Figure 4 shows external and internal photographs of the prototype hardware. The internal photograph is viewed from the lower-right corner of the hardware part of Figure 3.

The prototype hardware consists of the following three components, as shown in Figure 5:

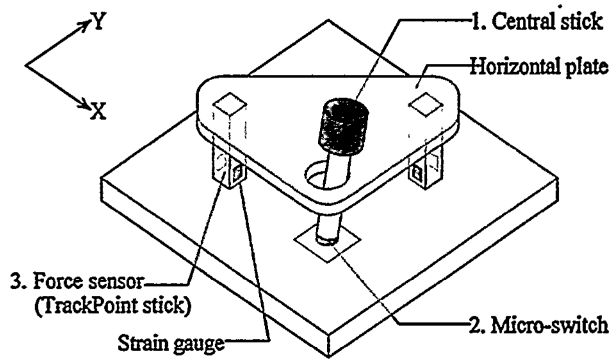

Figure 5. Structure of the prototype hardware

1. A central stick and a suspension mechanism[4] that moves the stick back to its neutral position
2. A micro-switch, located under the central stick, that is used for the click operation
3. Force sensors that detect a two-dimensional force exerted in the horizontal plane by the central stick

For this prototype, we used the force-sensing mechanism of the TrackPoint, which is composed of a stick and a special processing circuit.

At the bottom end of the TrackPoint stick, there are four strain gauges to detect a two-dimensional force exerted on the stick. The values of the force are amplified and converted into digital signals by A/D. They are then translated into corresponding velocity values and output as a standard mouse interface signal by the special processing circuit.

We use two TrackPoint sticks and one special processing circuit, as shown in the left part of Figure 3. The central stick goes through the center of a horizontal plate, and two TrackPoint sticks parallel to the central stick are fixed to the plate at their top ends so that the lines connecting each of them to the central stick are orthogonal to each other in the plate. This horizontal plate corresponds to the ring in Figure 2. When the central stick is tilted and pushes against the horizontal plate, the exerted force is transferred to the TrackPoint sticks. Each of the sticks detects one element of a 2D orthogonal force. For example, when the central stick is tilted along the X-axis, a bending moment is generated at the bottom of one TrackPoint stick, while a torsional moment is generated at the bottom of the other stick. In this prototype, only the bending moment is used for force detection.

The central stick and the horizontal plate are loosely coupled with a considerable area of slack. This creates a force-insensitive zone around the neutral position of the stick, which avoids undesired detection of force when the stick is depressed for click input. It also provides the user with a sense of contact when the central stick is tilted from the neutral position and presses against the horizontal plate.

[4] The suspension mechanism is omitted in Figure 5.

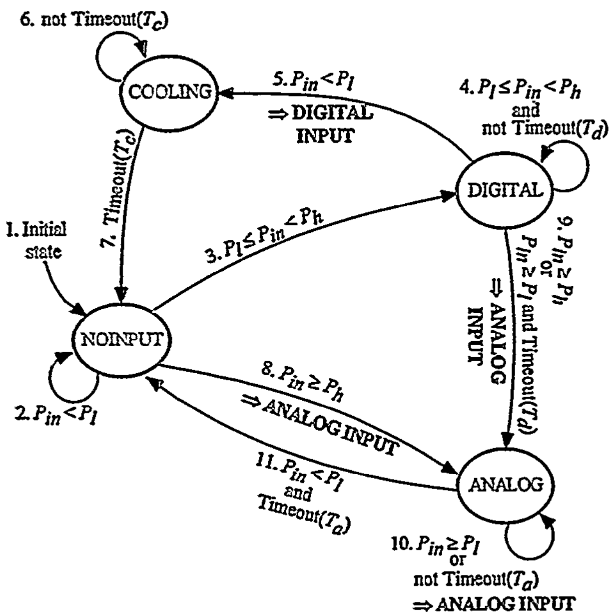

Figure 6. Software state machine

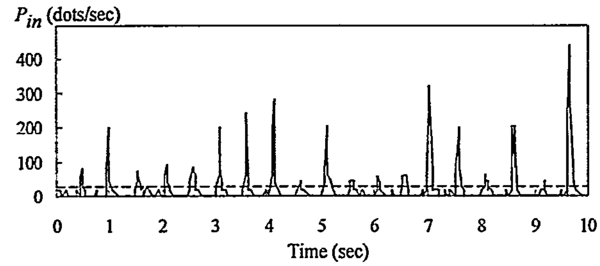

Figure 7. Results of the parameter-tuning experiment

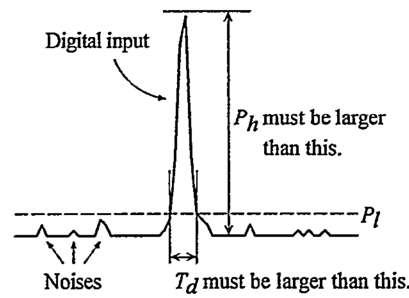

Figure 8. Parameters for discriminating a digital input

Software

In the prototype, the software discriminates digital and analog parts of the mouse-signal input from the hardware. This software is constructed as a state machine whose inputs are the velocity of the mouse (P_{in}) and timer information for timeout processing. It has the following four states:

NOINPUT — Initial state, represents no input

DIGITAL — State preparing a "digital input"

ANALOG — State processing "analog inputs"

COOLING — State to avoid chattering

Figure 6 shows transitions among states. In the figure, P_l, P_h, T_d, T_c, and T_a are threshold parameters for controlling the transition. Expressions shown along the transition arrows are conditions for the corresponding transitions, and statements after " ⇒ " indicate the handled (discriminated) inputs (digital or analog input). As for click input, input from the micro-switch, which corresponds to the left mouse button signal, is directly used.

Calibration

As explained in the previous subsection, the prototype software uses five constants, P_l, P_h, T_d, T_c, and T_a, as threshold parameters for state transition. These parameters can be calibrated by simple experiments.

The first experiment decides P_l, P_h, and T_d, which are parameters for discriminating the digital input. In this experiment, a user tilts the central stick until it touches the horizontal plate and returns the stick repeatedly. This operation corresponds to digital input. Figure 7 gives the results for one subject, K, and shows the temporal variation of the input P_{in} with 19 digital-input operations at intervals of about 0.5 seconds. Because of the implementation of the prototype hardware, the input P_{in} does not directly represent the force exerted on the plate, but rather the "velocity of the mouse" translated from the force information by the TrackPoint circuit. Therefore, the unit of the vertical axis of this graph is "number of mouse-pointer movements normalized to the value in 1 second (dots/sec)."

P_l is a parameter for canceling minute noises contained in the input P_{in}. In Figure 7, noises that are not intentional inputs by the subject exist at around 0, 0.8, and 6.3 seconds, etc. In the prototype, P_l is set to 30 dots/sec to cancel these noises. In the figure, the horizontal dotted line represents the threshold, and inputs smaller than this are ignored by the software.

P_h and T_d are central parameters for discriminating the digital input. A sequence of input that is smaller than P_h and continues no longer than T_d is handled as a digital input. In other words, P_h and T_d must be large enough to include all input sequences for which the user intends to perform digital inputs, as shown in Figure 8. From the results shown in Figure 7, P_h must be larger than about 500 dots/sec and T_d must be larger than about 200 msec for this subject. These parameters should essentially be adjustable according to the user. However, in the current implementation, fixed values $P_h = 800$ dots/sec and $T_d = 300$ msec are set as maximum-common values, on the basis of experiments with multiple subjects.

T_c is a parameter to avoid chattering just after a digital input has been processed, and inputs during this period are ignored. T_a is a parameter to prevent analog input from being cut into pieces by a momentary non-input status, and the analog-input state continues during this period even if the input is zero. These two parameters are introduced for comfortable operation. If they are too large, however, a user's intentional series of operations will be ignored. In the current implementation, both T_c and T_a are set to 100 msec.

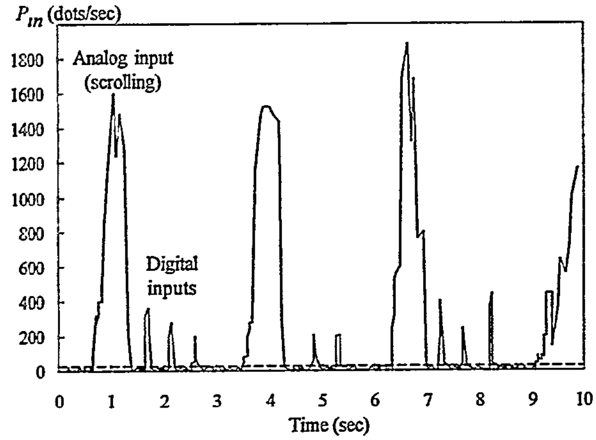

Figure 9. Actual input pattern for hypermedia browsing

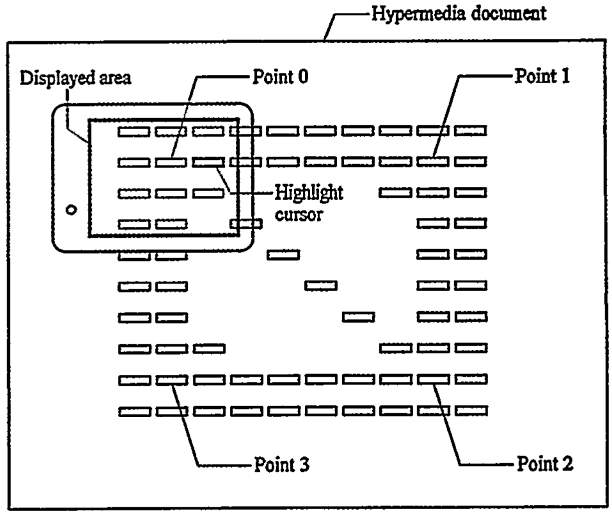

Figure 10. Hypermedia document for the evaluation

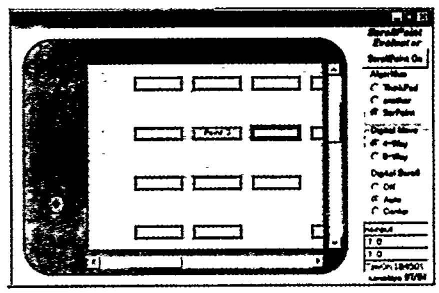

Figure 11. Screen of the evaluation program

Figure 9 shows the variation of the input P_{in} while a user performs hypermedia browsing using NaviPoint according to the method described in the previous section.[5] This graph illustrates very well the situation in which a user roughly scrolls a document by means of an analog input and then changes the highlighted item by means of several digital inputs. In this result, the intervals between input operations are at least 200 msec. Therefore, the value of 100 msec for T_c and T_a is considered to be reasonable.

EVALUATION EXPERIMENTS

Next, to verify the efficiency of NaviPoint in mobile information browsing, we carried out evaluation experiments. As a basis for comparison, we chose the "mouse and scroll bar" method (method A) which is the most widely used operation method. The time taken with NaviPoint was compared with the "mouse time" in several cases where both scrolling and pointing are used.

Experimental Procedure

The experimental procedure was as follows. Figure 10 shows an imaginary hypermedia document used in the experiments. Rectangles in the document represent clickable points (hyperlinks). The document is larger than the screen (16 times the screen size), and a user cannot view the whole of the document at one time. In such an environment, the following operation:

Exp. 1: Click point 0 → Scroll → Click point 1

Exp. 2: Click point 0 → Scroll → Click point 2

was performed repeatedly by subjects with a mouse or NaviPoint, and the elapsed times were measured.[6]

When a mouse was used, the scrolling operation was performed by dragging the slider of a scroll bar with the mouse pointer. Because scrolling at an angle is impossible with a scroll bar, two scroll bars must be operated successively in Exp. 2. When NaviPoint was used, the user had to first roughly scroll to the neighbor of the target item by means of an analog input, then adjust the highlight to the target item with several digital inputs, and select the item with a click input.

The subjects in the experiments were nine computer users who had mastered mouse operation. They received about 10 to 20 minutes advance explanation and training in operating NaviPoint. The NaviPoint prototype is held with two hands and its stick is manipulated with the left thumb.

The experiments were carried out on an IBM ThinkPad 755C (9545-L, Intel DX4-75MHz). The operating system was Windows 95, and the speed of the mouse pointer was set to medium. Figure 11 shows the display of the evaluation program. The screen size of the imitated mobile device is 320 x 240 dots (105 x 79 mm in absolute size) and the size of each clickable item is 64 x 16 dots (21 x 5 mm in absolute size).

Results and Discussion

Figure 12 shows the results of the experiments. These graphs show the elapsed times for nine subjects with a mouse and NaviPoint. Each bar shows the average elapsed time of ten examinations, and the supplementary vertical line shows the 95% confidence interval. The number over the initial of each subject represents the ratio of the NaviPoint time to the mouse time.

[5] More precisely speaking, an evaluation program that will be described in the next section is used.

[6] The main object of the experiments was to measure the performance of scrolling and pointing. Therefore, when a clickable point was selected, only a beep sound was processed, and the document was not replaced.

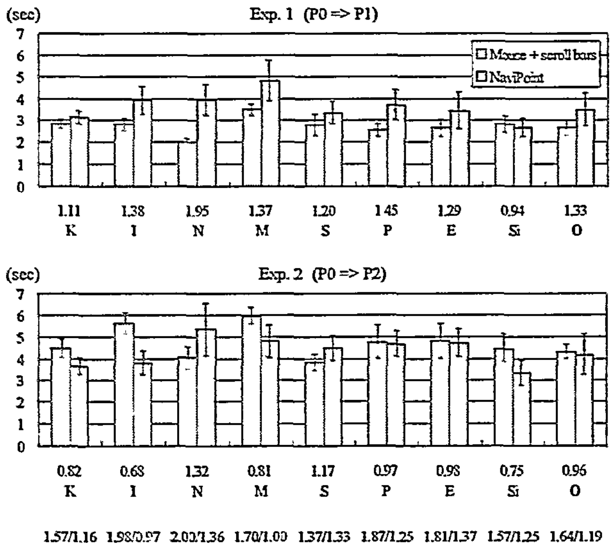

Figure 12. Evaluation result
(comparing mouse and NaviPoint)

From the result of Exp. 1, it can be said that NaviPoint can be manipulated with an overhead of less than 50% on the usual mouse operation.[7] Considering the advantages of NaviPoint in a mobile environment, this overhead is acceptable for practical use. However, the deviation with NaviPoint tends to become larger than with a mouse. Possible reasons for this phenomenon are as follows:

- The number of digital inputs affects the elapsed time.
- Faulty execution of digital and analog input causes unintentional movement and increases the elapsed time.

The latter is thought to be the major reason, since the difference is very small for subjects K and Si, who mastered NaviPoint very well. In this regard, several improvements are being planned for more accurate operation, such as calibrating the parameters dynamically for each user, as explained in the previous section, and redesigning the physical hardware.

In Exp. 2, many subjects achieved better results with NaviPoint, because of its advantage of allowing the user to scroll in any direction. The number at the bottom of Figure 12 shows the ratio of the result in Exp. 2 to Exp. 1 for each subject and operation method. Compared to the 1.4–2 times increase in mouse operation, NaviPoint shows an increase of less than 1.4 times for all subjects. Considering the larger scrolling distance in Exp. 2, it can be said that with NaviPoint a user can operate with the same performance and feeling in any direction.

To summarize the results, NaviPoint is suitable for browsing hypermedia documents with arbitrary scrolling on a small screen in a mobile environment.

[7] As an exception, subject N showed a 95% overhead. However, this can be considered as a special case in which the user had mastered mouse operation extremely well.

RELATED WORK

As a result of the spread of the WWW, the importance of the scrolling operation has received greater attention recently. For example, Microsoft's IntelliMouse [10] can perform scrolling (or zooming) by rotating a wheel installed between two mouse buttons. Zhai et al. have proposed several methods that combine scrolling with pointing — such as using a mouse with a TrackPoint device, and using multiple pointing devices — and compared and examined them [21]. These devices and ideas are mainly concerned with the desktop environment, and try to add a direct scrolling operation by adding new parts without losing the feel of the original operation.

As input devices for mobile information devices, pens and keypads (cursor keys) are currently popular. The problems associated with these devices in information browsing have already been discussed in the early section of this paper. Other devices for a mobile environment include a depressable digital dial [18, 20]. The functions of this device are almost the same as those of cursor keys, but it has the advantage of being manipulatable with one hand. However, arbitrary scrolling is difficult, because it allows input in only one dimension.

Among the new scrolling methods for mobile information devices is the "Scroll Display" proposed by Siio [19]. This is a display device, with a built-in mouse on the back, in which the movement of the display itself is treated as a scrolling operation. This idea may be effective as a metaphor for intuitive scrolling operation, but the device is difficult to use while moving. As a method for improving scrolling efficiency, "Alphaslider" has been proposed by Ahlberg et al. [1]. It is effective for selecting an item from a text-based list or menu, but its performance in scrolling of general graphics is considered to be the same as that of a normal scroll bar.

Other interesting input devices include "Rockin'Mouse" [2] and Rekimoto's tilting display [14]. These devices utilize "tilting" operation as an additional input method, and are very useful for controlling 3D objects interactively. However, they are not always suitable for mobile information browsing, because the first needs a flat surface to operate on and the second requires the display itself to be moved.

Many input devices that can be used with one hand or without any hands have also been proposed [4, 9, 17], but most of them need special dexterity for operation and/or are difficult to integrate into a mobile information device such as a PDA.

The most notable characteristic of NaviPoint in relation to the above devices and methods is that it is a specialized device for mobile information browsing. By sacrificing the generality of mouse pointer display and character input, NaviPoint gains an important advantage for mobile information browsing in that it can be manipulated with just one finger without watching.

CONCLUSION

This paper has reported a new input device, NaviPoint, that allows hypermedia information to be browsed comfortably on mobile information devices such as PDAs. With this device, three types of input — "analog input," "digital input," and "click input" — can be performed by using just one finger. Experiments using the prototype have shown that user's performance in hypermedia browsing with NaviPoint was comparable with their performance when they used existing operation methods.

Although they are not mentioned in this paper, NaviPoint has other advantageous features for the mobile environment, such as easy waterproofing [6]. It can also be used to control other devices such as a television set [15]. However, NaviPoint is a specialized device for information browsing, and is not suited for inputting information such as characters. Therefore, to build up a real mobile information device, it must be combined with another input device such as a pen.

One future task will be to build and evaluate such a mobile information device that incorporates NaviPoint. Another theme at the software level is the construction of an operation scheme suitable for information browsing that incorporates menu handling. As another idea, we are currently planning to apply a software part that implements input discrimination in existing hardware such as TrackPoint. Investigation of the microstructure of the movement [11] and evaluation on the basis of Fitts' law [3] are also planned.

REFERENCES

1. Ahlberg, C. and Shneiderman, B. The Alphaslider: A Compact and Rapid Selector, in *Proceedings of ACM CHI '94 Conference*, 365–371, 1994.
2. Balakrishnan, R. et al. The Rockin'Mouse: Integral 3D Manipulation on a Plane, in *Proceedings of ACM CHI '97 Conference*, 311–318, 1997.
3. Fitts, P.M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology 47*, 6, 381–391, 1954.
4. Fukumoto, M. and Tonomura, Y. "Body Coupled FingeRing": Wireless Wearable Keyboard, in *Proceedings of ACM CHI '97 Conference*, 147–154, 1997.
5. Kawachiya, K. and Siio, I. A Framework for Mobile Information Cache, in *Proceedings of 53rd Annual Convention IPS Japan*, 1B-9, 1-17–1-18, in Japanese, 1996.
6. Kawachiya, K. and Ishikawa, H. A Single-Finger Controller for Mobile Devices, in *Proceedings of 54th Annual Convention IPS Japan*, 4R-3, 4-111–4-112, in Japanese, 1997.
7. Kawachiya, K. and Ishikawa, H. A Useful Input Device for Hypermedia Browsing. *IPSJ SIG Notes 97*, 43 (97-HI-72), 55–60, in Japanese, 1997.
8. Kawachiya, K. and Ishikawa, H. ScrollPoint: An Input Device for Mobile Information Browsing. IBM Research Report, RT0211, 1997.
9. McAlindon, P.J. et al. The Keybowl: An Ergonomically Designed Document Processing Device, in *Proceedings of ACM ASSETS '96*, 86–93, 1996.
10. Microsoft Corp. Microsoft IntelliMouse Home Page. http://www.microsoft.com/products/hardware/intellimouse/default.htm.
11. Mithal, A.K. and Douglas, S.A. Differences in Movement Microstructure of the Mouse and the Finger-Controlled Isometric Joystick, in *Proceedings of ACM CHI '96 Conference*, 300–307, 1996.
12. Narayanaswamy, S. et al. Application and Network Support for InfoPad. *IEEE Personal Communications 3*, 2, 4–17, 1996.
13. Negishi, Y. et al. Tuplink: A System Structure for Mobile Micro Clients. IBM Research Report, RT5123, 1997.
14. Rekimoto, J. Tilting Operations for Small Screen Interfaces, in *Proceedings of ACM UIST '96*, 167–168, 1996.
15. Robertson, S. et al. Dual Device User Interface Design: PDAs and Interactive Television, in *Proceedings of ACM CHI '96 Conference*, 79–86, 1996.
16. Rutledge, J.D. and Selker, T. Force-to-Motion Functions for Pointing, in *Proceedings of INTERACT '90: The IFIP Conference on Human-Computer Interaction*, 701–705, 1990.
17. Salem, C. and Zhai, S. An Isometric Tongue Pointing Device, in *Proceedings of ACM CHI '97 Conference*, 538–539, 1997.
18. Siio, I. and Murata, H. A palmtop computer equipping a dial switch, in *Proceedings of. 54th Annual Convention IPS Japan*, 4R-1, 4-109–4-110, in Japanese, 1997.
19. Siio, I. Scroll Display: Pointing Device for Palmtop Computers. *IPSJ SIG Notes 97*, 24 (97-HI-71), 91–98, in Japanese, 1997.
20. Sony Electronics Inc. Sony Simplifies Wireless Communications With Unique Jog Dial Control. http://www.sel.sony.com/SEL/corpcomm/news/wtc/6.html.
21. Zhai, S., Smith, B.A., and Selker, T. Improving Browsing Performance: A Study of Four Input Devices for Scrolling and Pointing Tasks, in *Proceedings of INTERACT '97: The 6th IFIP Conference on Human-Computer Interaction*, 286–292, 1997.