

Title	Design and implementation of the double mouse system for a window environment
Author(s)	Nakamura, Satoshi; 塚本, 昌彦; 西尾, 章治郎
Citation	IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings. 1 P.204-P.207
Issue Date	2001-08
Text Version	publisher
URL	http://hdl.handle.net/11094/14053
DOI	
rights	c2001 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE..
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

Design and Implementation of the Double Mouse System for a Window Environment

Satoshi Nakamura
Graduate School of Engineering,
Osaka University
nakamura@ise.eng.osaka-u.ac.jp

Masahiko Tsukamoto
Graduate School of Engineering,
Osaka University
tuka@ise.eng.osaka-u.ac.jp

Shojiro Nishio
Graduate School of Engineering,
Osaka University
nishio@ise.eng.osaka-u.ac.jp

Abstract

Recently, due to remarkable advancements of computer technology, multiple applications can be simultaneously run on a single computer. These applications commonly use a mouse as an input device. However, with a single mouse, we can only input two-dimensional coordinates and press mouse buttons for input. Therefore, this conventional input method using a single mouse handled by one hand is not sufficient to support complex or parallel operations in sophisticated applications. On the other hand, it is widely recognized that using two different input devices handled by two hands is useful for complex operations. In this paper, we show our design and implementation of a multipurpose interface which integrates two mice as input device for window system.

1. Introduction

Recently, due to remarkable advancements of computer technology, multiple tasks can be simultaneously run on a single computer. For example, when we create documents with a word processor, we can insert figures and tables created by a draw tool or spreadsheet tool into a document. In addition, due to advancement of software technology, we can create and play music and computer animations on a computer. The computer has become a popular tool for a wide range of purposes; however, these tools commonly use a mouse as an input device, we can therefore only input two-dimensional coordinates and press mouse buttons with a single mouse. Therefore, to support complex or parallel operations in such sophisticated applications, a single

mouse handled by one hand is not sufficient as an input device.

On the other hand, it has been widely recognized that using two different input devices handled by two hands is useful for complex operations. In fact, there were several studies on such two-handed inputs in a draw tool and a map navigation tool. These studies have shown that the use of two-handed input for complex operations is effective. However, the developed systems and the proposed methods are specific to a particular application. In this paper, we show our design and implementation of a multipurpose interface which supports two mice as input devices for a window system such as Microsoft Windows and X-Windows. We call this system the DoubleMouse system. In this paper, first, we briefly describe related work. Next, we propose the DoubleMouse system, and we explain our implementation of the DoubleMouse system. In the last section, we present our conclusions and plans for future work.

2. Related Work

It has been widely recognized that using two different input devices handled by two hands is useful for complex operations [4, 6]. In fact, there were several studies on such two-handed inputs in a draw tool with a pen and a trackball [2, 9], a map navigation tool with a pen and a puck [5], and a user interface for three-dimensional neurosurgical visualization with a miniature head, a cutting-plane, and a stylus [7]. An object is controlled in three-dimensional space with two-handed manipulation [1]. These studies have shown the effectiveness of two-handed input for complex operations. In addition, experimental examinations had shown the effectiveness of two-handed input for complex opera-

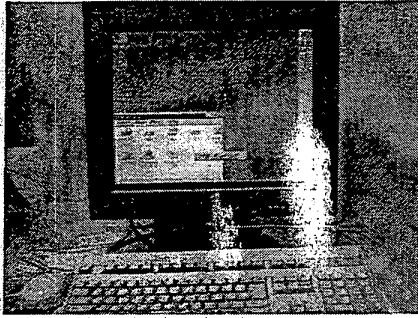


Figure 1. A system environment example of using the DoubleMouse system.

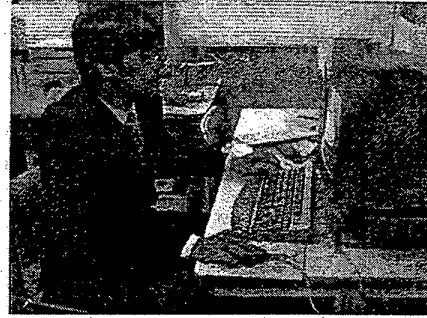


Figure 3. Manipulating a window system with the DoubleMouse system.

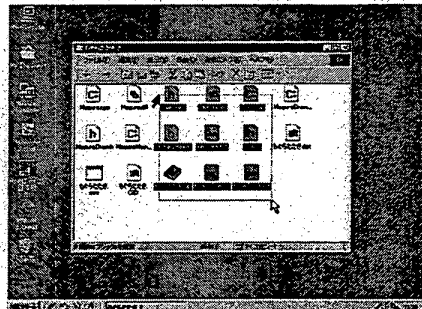


Figure 2. A screen snapshot example of using the DoubleMouse system.

tions [10].

On the other hand, a manipulating system [3] uses a PDA (Personal Digital Assistant) and a mouse in order to offer good performance for some window control for users. For example, users can perform page down, page up, go home, go end and etc, for scrolling web pages by only touching a button on a PDA. However, this system supplies only good window control, but not good window system control.

3. DoubleMouse System

In the DoubleMouse system, two mice, called the right mouse and the left mouse, are connected to a computer as input devices (Figure 1), and both the right mouse cursor and the left mouse cursor appear on a window screen (Figure 2).

The DoubleMouse system provides the following functionality in operating a window system:

Mouse movement reduction: Due to the increasingly

complexity of applications, the number of steps and manipulations to perform a task has increased. In addition, the number of menus, buttons, window movements and the accumulation of mouse movement needed increased the overall amount of needed mouse movement. This accumulation of mouse movement can be decreased by introducing the DoubleMouse system into these systems because we can then select the mouse cursor with a shorter distance to the target position.

Simultaneous indication of two points: A window system has many tasks where we have to indicate two points like dragging, dropping, and selecting an area. If we indicate two points with a single mouse on a computer, we have to combine mouse movements and button controls. Therefore they are not straightforward operations. If we apply the DoubleMouse system to control a window system, we can then indicate two points at a time with two cursors. As a result we efficiently carry out operations.

Window movement: In order to change the location of a window in a conventional environment, we have to select the title bar at the upper edge of the window and then drag the mouse. This manipulation requires sensitive and complex control with the mouse. Instead, by using the whole window as the title bar when the left mouse is used, we can reduce the required sensitivity. Therefore, we can control the window more easily in our system. This function is well suited for the left mouse (for left-handers, it is visa versa) because most people cannot do extremely sensitive manipulation with their left hand. With this window movement feature, we may be able to do two-handed interaction effectually with two mice on a screen.

Window focus change: On a window system, we typically use many windows for many tasks. In order to

switch tasks we have to click on a window which may be overlapped by another window. To make this job easier, we use the wheel of a wheel mouse as a task switcher, similarly to the jog dial function of a cellular phone. First, we add the window to a list of current windows. If we would like to change the window in front of the list, we rotate the wheel and select a window from the window list. When we need to add or delete a window from the window list, we click the wheel button on a window. This function is called the "Jog Menu" function.

Application specific manipulation: The DoubleMouse system becomes more powerful applications are aware of its functions. For example, manipulations of a draw tool with the DoubleMouse system allow a user to move a palette window by the left hand and to draw by the right hand. In the real world, when we change colors while drawing, we often move a palette nearby the brush and then change colors. This method allows a user to do a similar manipulation on the computer.

4. Implementation

4.1. Base technology

In order to realize the DoubleMouse system, two mice should be enabled by a computer and their cursors should be manipulated on a computer.

Two mice connection: The computer should enable two mice as input devices.

Independent control of two mice: Even if a computer enables two mice, conventional OS (Operating System) cannot identify these inputs. Therefore, an independent control mechanism for these two inputs should be realized.

Reflection of mouse input: There are no functions in a conventional operating system to display two mouse cursors on a screen and to give distinctly orders to several inputs in the OS. Thus, we should provide a mechanism to reflect two mice inputs properly.

Our solution to realize these mechanisms is as follows:

Using both serial and PS/2 ports: The serial port and the PS/2 port in a computer are used to support two mice at a time.

Direct control of the serial port: The PS/2 mouse is enabled by the usual OS drivers. The serial mouse is controlled directly by the DoubleMouse driver. The serial port is monitored to analyze data from the mouse to recognize mouse movements and button actions.

Emulation of the left mouse cursor: We display the additional mouse cursor on the screen and emulate the left mouse functions such as button down and double

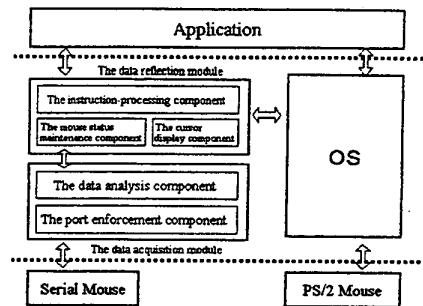


Figure 4. The DoubleMouse system setup.

click operations. Therefore, we send its behaviors to OS through the mouse API (Application Program Interface).

4.2. A system configuration

Figure 4 shows the system setup of the DoubleMouse system. The right mouse is connected via a PS/2 port and the left mouse is connected via a serial port of the computer. The right mouse action data is transmitted from the PS/2 port through the OS's drivers, and the right mouse behavior is reflected on a computer as message events. The data received from the serial port comes directly from the left mouse. By analyzing this input data, this system can know the mouse action being performed. It then displays the cursors on the screen and emulates the appropriate mouse behaviors based on their data.

The DoubleMouse system is composed of the following two modules:

- The data acquisition module
- The data reflection module

We will describe of these modules in more detail the rest of this section. The data acquisition module This module reads the serial port data and analyzes the data. It then sends messages to the module of reflection of mouse data.

This module is composed of the following two components:

The port enforcement component: This component listens on the serial port connected to the serial mouse. If the port receives data, this component sends the data to the data analysis component.

The data analysis component: This division analyzes mouse behavior data received from the port enforcement component; it monitors the behavior of the mouse such as mouse movement and button actions. It then accordingly, sends mouse behavior messages to the mouse status maintenance component.

4.2.1 The data reflection module

This module updates the mouse status such as the mouse position and mouse button status based on the received data. In addition, it displays the left mouse cursor on a screen and sends messages to the system in order to emulate the left mouse behavior.

This module is made with the following three components:

The mouse status maintenance component: When this component receives data from the data acquisition module, it updates the mouse status based on the received data. In addition, this component sends then mouse position to the cursor drawing component and then sends mouse behavior to the instruction-processing component.

The cursor display component: This component draws the left mouse cursor on a screen based on received mouse position data.

The instruction-processing component: This component sends mouse behavior to those affected windows.

4.3. DoubleMouse Library

In order to allow applications to directly use features of the DoubleMouse system, we implemented the DoubleMouse system library. We designed the DoubleMouse system API similarly to the mouse API of MS Windows so programmers could easily understand it.

The APIs consist of the following functions:

- Acquisition and setting of a position of mouse cursor.
- The functions of mouse button switch.
- Acquisition and setting of a capture window.
- Setting of a mouse cursor mode such as show/hide.
- Acquisition and setting of cursor movable area.

The DoubleMouse system always listens to the left mouse and shares its status with the DoubleMouse DLL (Dynamic Link Library). Applications can get and set the mouse status by using this library.

The refresh rate of the left mouse is set out 20 frames per second. We have confirmed that Windows works without any conspicuous performance degradation on an MMX Pentium 233 MHz PC with the DoubleMouse system.

5. Conclusion

In this paper, we proposed the DoubleMouse system as manipulation mechanisms for a window system and show our design and implementation of a prototype system.

We are continuing our work on the concepts surrounding DoubleMouse system. We are implementing the functions to get/set the mouse cursor and ones to get/set the double click interval of the left mouse. We are evaluating

operationality with the DoubleMouse system. We are designing the USB (universal serial bus) version, and three or four mice version, and implementing a manipulation system with both hands and both legs [8]. In addition, we will apply the DoubleMouse system to three-dimensional operations [7, 11].

Acknowledgments This work is supported by Research for the Future Program of Japan Society for the Promotion of Science under the Project "Researches on Advanced Multimedia Contents Processing"(JSPS-RFTF97P00501).

References

- [1] Balakrishnan, R., Kurtenbach, G.: Exploring bimanual camera control and object manipulation in 3D graphics interfaces; *Proceeding of the CHI 99 conference on Human factors in computing systems*, pp. 56–62, 1999.
- [2] Bier, E., Stone, M., Pier, K., Buxton, W., DeRose, T.: Toolglass and magic lenses: the see-through interface; *Proceedings of the 20th annual conference on Computer graphics*, pp. 73–80, 1993.
- [3] Brad A. M., Kin Pou Lie, Bo-Chieh Yang.: Two-handed input using a PDA and a mouse; *Proceeding of the CHI 2000 conference on Human factors in computing systems*, pp. 41–48, 2000.
- [4] Buxton, W., Myers, B.: A study in two-handed input; *Conference proceedings on Human factors in computing systems*, pp. 321–326, 1986.
- [5] Hinckley, K., Czerwinski, M., Sinclair, M.: Interaction and modeling techniques for desktop two-handed input; *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pp. 49–58, 1998.
- [6] Hinckley, K., Pausch, R., Goble, J., Kassell, N.: A survey of design issues in spatial input; *Proceedings of the ACM symposium on User interface software and technology*, pp. 213–222, 1994.
- [7] Hinckley, K., Pausch, R., Proffitt, D., Kassell, N.: Two-handed virtual manipulation; *ACM Transactions on Computer-Human Interaction*, Vol.5, No.3, pp. 260–302, 1998.
- [8] HUNTER DIGITAL: Nohand mouse; <http://www.footmouse.com/>.
- [9] Kurtenbach, G., Fitzmaurice, G., Baudel, T., Buxton, B.: The design of a GUI paradigm based on tablets, two-hands, and transparency; *Conference Proceedings on Human factors in computing systems*, pp. 35–42, 1997.
- [10] Leganchuk, A., Zhai, S., Buxton, W.: Manual and cognitive benefits of two-handed input: and experimental study; *ACM Transactions on Computer-Human Interaction*, Vol.5, No.4, pp. 326–359, 1999.
- [11] Zeleznik, R., Forsberg, A., Strauss, P.: Two pointer input for 3D interaction; *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pp. 115–120, 1997.