# ThumbSense: Automatic Input Mode Sensing for Touchpad-based Interactions

**Jun Rekimoto**
Interaction Laboratory
Sony Computer Science Laboratories, Inc.
3-14-13 Higashigotanda Shinagawa-ku, Tokyo 141-0022, Japan
Phone: +81 3 5448 4380 Fax: +81 3 5448 4273
Mail: rekimoto@acm.org
http://www.csl.sony.co.jp/person/rekimoto.html

## ABSTRACT

While manipulating the touchpad, a user's hand position must be away from the keyboard's home position. This effect hinders smooth switching between text entry and pointer manipulation, and is considered to be the one of the major drawback of the touchpad against to the trackpoint. This paper introduces ThumbSense, a new input technique aims to solve this problem by automatically sensing users' input mode based on finger contact to the touchpad. A key on the keyboard, such as the F key, transparently acts both as a normal key as well as a mouse button. This technique is implemented by using the sensor feature of the touchpad, and is possible to apply most of currently available portable computers without requiring any additional hardware/sensors.

### Keywords

Touchpad, touch-sensitive user interfaces, mode sensing.

## INTRODUCTION

*Touchpad* is one of the most popular input devices for portable computers. This device detects finger position based on capacitive sensing and can be used as a mouse compatible input device with the combination of attached buttons. However, in contrast to Trackpoint, users of touchpad are forces to change their hand positions from the keyboard home position to the position of the touchpad (Figure 1). Although it is not impossible to operate the touchpad from the keyboard home position by stretching thumbs toward the touchpad buttons, it is neither comfortable nor natural. In contrast, Trackpoint is designed to be operated while the user's hands are on the home position. Because of this difference, users who are capable of touch-typing tend to prefer trackpoint to touchpad.

This paper proposes a new interaction technique, called *ThumbSense*, to allow touchpad users to manipulate it while keeping the keyboard home position. The primary idea is to use keyboard keys, such as F or J keys, as mouse buttons. The ThumbSense software automatically distinguishes user's input modes, typing and mouse operation modes, by sensing whether the user's thumb is contacting the touchpad surface.

Copyright is held by the author/owner(s).
*CHI 2003*, April 5–10, 2003, Ft. Lauderdale, Florida, USA.
ACM 1-58113-637-4/02/0004.

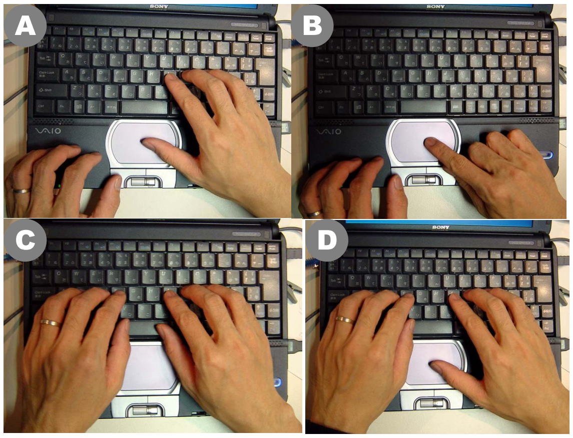

**Figure 1: Comparison of hand positions (A, B: Without ThumbSense. The user's hands are off from the home position. C, D: with ThumbSense. The user can transparently switch between the text (C) and mouse (D) modes.)**

### ThumbSense

The touchpad senses the users finger position based on capacitive sensing. It also detects finger contact to the touchpad, and thus it can be used as a kind of touch sensors, but this feature has not been effectively used in previous user interface systems. The ThumbSense recognizes user input modes using this touch sensitivity. When a user's finger (thumb) touches the touchpad, the system automatically becomes the "touch-mode" and preocesses keyboard keys. For example, the F and J keys act as mouse buttons. With this feature, users of ThumbSense can operate the touchpad without moving their hands away from the home position. To enter characters, users can simply lift their thumb from the touchpad, and F and J keys return to the normal keys. This touch-sensing feature provides transparent mode switching between the key-input and the mouse-operation modes without requiring explicit mode change commands. The mouse "wheel" button can also be emulated; moving the thumb on the touchpad while pressing the S key would cause wheel motion events.

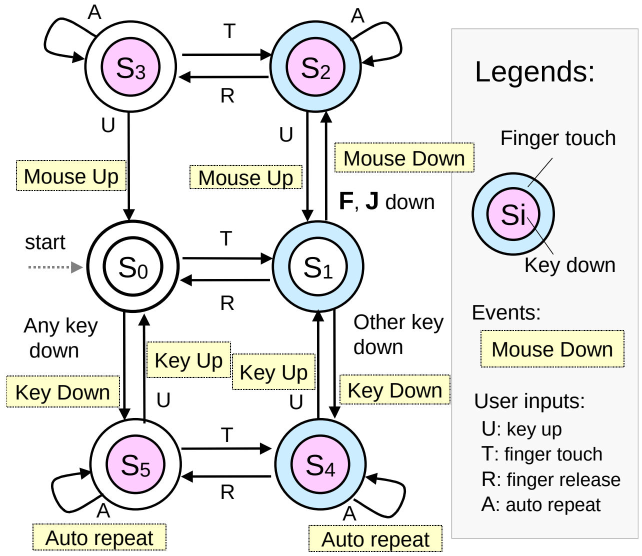

**Figure 2: The state transition diagram.**

Figure 2 shows the state transition diagram of ThumbSense for emulating the mouse. Basically, the system distinguishes the key-input and the mouse modes based on finger contact. However, during the ”drag” operation, the finger is often lifted for repositioning (because of the limited area of the touchpad), and thus the user’s finger is temporarily away from the touchpad. To consider this situation, the system enters the ”drag” mode when the user first presses the emulated mouse buttons (i.e., corresponding keyboard keys). While the user holds the mouse buttons, the input mode is fixed to the mouse emulation mode regardless of the finger contact to the touchpad.

## INTERACTION WITH MULTIPLE-BUTTON MOUSE

As described in the previous section, ThumbSense emulates the normal mouse device with three buttons (the third button could be a wheel). Since the keyboard has more than three keys, it has a potential to design interaction techniques that has multiple buttons.

**Symmetric mouse button assignment**: We can define $\boxed{F}$, $\boxed{J}$ as the left mouse button, $\boxed{D}$, $\boxed{K}$ as the right mouse button, and $\boxed{S}$, $\boxed{L}$ as the wheel button. This symmetric assignment enables flexible mouse operations with either by one hand or two hands (e.g., both mouse buttons and the touchpad can be operated by the right hand).

**As a new modifier key**: ”Touch” can be regarded as a new modifier key. For example, a user could launch the mail application by touch-$\boxed{M}$ (pressing the $\boxed{M}$ key while touching the keypad), or change the window order by touch-$\boxed{E}$, etc. These key operations can be followed by mouse operations. Within a drawing application, a user could directly create a rectangle by touch-$\boxed{R}$ + dragging (i.e., without selecting a ”rect” tool). Another example is also shown in Figure 3.

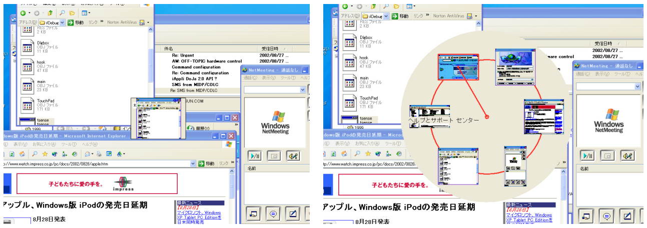

**Figure 3: An example of multi-button mouse operations. One of the keyboard keys is bound to the application switch function. By simply typing this key, it acts as the normal task-switching key (i.e., ALT-TAB of Windows), but sliding the thumb (finger) on the keypad, the task selection pie menu appears on the screen.**

**Enhance mouse operations**: While performing a drag operation, it is often the case that we want to drop an object to the window that is obscured by another window. With ThumbSense, dragging can be modified by pressing other keys. For example, a user can start dragging operation by touch-$\boxed{F}$, and change the window order by pressing $\boxed{E}$ without releasing the $\boxed{F}$ button.

## IMPLEMENTATION

The ThumbSens system is implemented as a taskbar software module for Microsoft Windows.[1] It intercepts keyboard and mouse events, and generates and delivers appropriate events according to the state transition as shown in Figure 2. Since this event emulation occurs at the OS level, no modification is required to applications. The low-level APIs to the touchpad are also used to detect finger contact to the touchpad. Currently, two major touchpad devices (Synaptics and ALPS) are supported, and can be used with the most of the existing notebook computers.

## CONCLUSION

The ThumbSense is a new interaction technique that allows users to operate the touchpad input device without put hands aware from the keyboard home position. This feature solves one of the major drawbacks of the touchpad against to the trackpoint. ThumbSense can be regarded as a kind of sensory enhanced user interfaces [1, 2]. Unlike these previous systems, ThumbSense uses the feature of the existing input device, and thus can be widely used with currently available notebook computers without requiring any additional hardware or sensors.

## Acknowledgements

We thank Sony Corporation and ALPS electronics Inc. for providing us the touchpad driver information.

## REFERENCES

1. Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In *CHI’98 Proceedings*, pages 17–24, 1998.

2. Ken Hinckley and Mike Sinclair. Touch-sensing input devices. In *CHI’99 Proceedings*, pages 223–230, 1999.

[1] downloadable from http://www.csl.sony.co.jp/person/rekimoto/tsense/