DGTS: Integrated typing and pointing

DGTS: Integrated Typing and Pointing

Iman Habib[1], Niklas Berggren[1], Erik Rehn[1], Gustav Josefsson[1], Andreas Kunz[2] and Morten Fjeld[1]

Chalmers University of Technology, Rännvägen 68, SE 412 96 Gothenburg, Sweden
fjeld@cs.chalmers.se
[2]ETH Zurich, Tannenstrasse 3, CH-8092 Zurich, Switzerland
kunz@iwf.mavt.ethz.ch

Abstract. Capacitive sensing is used in many different fields of application. It has been implemented in such devices such as mobile phones and remote controls. However, up until now the physical sensing area has remained limited despite the widespread use of larger input devices such as keyboards. We present GFist, which seamlessly integrates keyboard typing and cursor pointing. This input device offers multi-finger operation for scrolling and other specialized input commands. The objective of this work is to replace computer mice and touchpads by integrating capacitive sensing into a layer within the keyboard. The advantage of this approach is the reduced space required for pointing devices in laptops. This paper introduces the technical background, describes our contribution, and concludes with primary functional tests.

Keywords: Capacitive sensing, touchpad, multi pointer, typing, pointing

1 Introduction

Ever since Graphical User Interfaces (GUIs) were developed for computers, the mouse has been the most common pointing device. The physical space occupied by a mouse is considerable. Most laptop computers come with a touchpad or a pointing stick. A pointing stick is an isometric joystick that is situated close to or in the middle of the keyboard and is controlled by the user's index finger. Capacitive sensing was introduced in PreSense [1] and SmartPad [2] as a technology to apply with a large sensor for controlling a mouse pointer on a mobile device. Another example is IMPAD, a paper-thin, pressure sensitive multitouch device [3]. Capacitive touch-pads have also been used in combination with a keyboard by ThumbSense [4] to change the keyboard's operation mode. Other input technology such as the layered touch panel [5] or the PointingKeyboard [6] employ additional infrared-grid sensors to realize more functionality. Most of these systems offer limited interaction space.

Integrating capacitive sensing into a keyboard presents at least two issues. Firstly, there is the height detection problem regarding *touchdown* and *liftoff;* that is, the sensors might have difficulties detecting whether or not a finger has touched the keyboard [7]. Secondly, there is the issue of the mechanics of the key's sensors. In order to detect whether a key is pushed, the capacitive sensors must not be in the way of the keyboard's regular circuitry, thus requiring a thin and flexible sensor surface.

2 Contribution

We present GFist (GF)[1], which employs capacitive sensors to detect a finger's position on/over the keyboard. Previous research has shown that capacitive finger sensing can be done in multiple ways, each with their respective pros and cons [8][9]. The different sensor modes also have an influence on the required electrode grid and thus on the available resolution. The tables presented below compare alternative sensor modes and patterns. The alternative sensor modes each have their pros and cons, and consequently they are used for different applications. Table 1 presents three alternative sensor modes with their pros (+), cons (–). Their areas of use are capacitive buttons and large multi-touch surfaces (shunt mode), multi-user systems (transmit mode), and touchpad and capacitive buttons (loading mode).

Table 1. Sensor modes with pros [+] (left) and cons [–] (right).

Shunt mode + Delimited area of sensitivity + Multiple senders and receivers + Electrode can have sender and receiver status	– Difficult to shield – 2 electrodes/sensors
Transmit mode + User-sensitive + 1 electrode/sensor	– Less quality in measurements with increasing user grounding
Loading mode + Easy to shield + 1 electrode/sensor	– Sensitive to all conductive objects – Short detection range

By combining information from several sensors, an image of hands and fingers on the keyboard can be created. Depending on the sensor patterns [10], the sensors create two-dimensional images with different characteristics. Table 2 shows a number of sensor patterns with pros (+), major-pros (+ +), cons (–), and other relevant facts.

Table 2. Comparison of sensor patterns, [–, +, + +].

Sensor Pattern	Surface/Electrode	Pattern Type	Sensor Modes	Multi-touch
CellMatrix	–	Matrix	Loading, Shunt	++
SnakePit	+	X-Y	Loading	–
Synaptics	+	X-Y	Loading	–
Hexagonal	+	X-Y	Shunt	–
TT [7], [11]	+	X-Y	Loading	–

Unlike SmartSkin [12], which works in shunt mode, GF is based on the CellMatrix pattern working in loading mode. We chose the loading mode principle together with the CellMatrix layout, since it suits best our requirements for simplicity of the setup and the precision of the position readout. We use 1 cm^2 sensors, average multiple

[1] Video presentation of GFist: http://www.t2i.se/pub/media/gfist.wmv

sensors' readouts, and achieve a resolution of ~1 mm combined with a high update rate for the sensor readout. The sensor size had to fulfill the following requirements:

- Since a finger's position is calculated from multiple sensor values, many sensors should be used for blob detection, i.e. they should be as small as possible.
- To realize a reasonable capacity and measure capacity changes when a finger approaches the sensor, the electrodes should be set to a size of a fingertip.

If the electrode's size is too large, the proximity of a finger only causes small capacity changes compared to the sensor's basic capacity. Thus, the change might not be detectable by the electronics (resolution of the A/D-converter). In our prototype a 96 cm^2 sensor surface with several functionalities was realized (Figure 1). It controls the cursor, detects when keys are pressed, recognizes simple gestures, and discerns when mouse clicking is performed on the keyboard's surface. The prototype's update frequency is 13.68 Hz. The data read is sent to the computer via a USB port.

Fig. 1. Left: Color value map and finger position on the keyboard. Right: GF electronics.

Electronics: The electrodes are connected to the capacitance-to-digital converter (CDC) from Analog Devices AD7147 via signal selectors [13]. The AD7147 consists of a sigma-delta-based CDC with 12 analog input channels and communicates with a microcontroller via an I^2C bus. Since 96 sensors exist so far, the 12 channels of the AD7147 are not enough. We use an 8 channel signal selector (analog multiplexer) to connect eight sensors to each of the CDC's 12 analog input ports. A microcontroller enables communication between the sensor board and the PC. The PC-software continually dispatches requests for data and then processes feedback.

Blob Detection: The blob detection algorithm starts with the sensor that has the highest readout. Next, all neighboring cells are assumed to belong to the same blob as long as they have a readout that is smaller than the previous one but still above a certain threshold (coming from the calibration). The algorithm stops if a neighboring cell's readout is below this threshold or if the readout increases again (Figure 1).

Software Design: The software performs several tasks. It handles incoming packets via USB and generates field images out of these packets. A three-step calibration consists of i) sampling background-noise level when the system is in idle mode, ii) setting a hi-pass filter by holding a finger fixed for a few seconds, and iii) normalization of a maximum sampling level by hovering a finger above the sensor area. Based on this, a sensor value map is provided showing detected fingers and is used to control the cursor (see Fig.1). In order to address existing applications like a browser, it is possible to use the sensor data to control a mouse pointer. Signal duration is used to distinguish between mouse movement (long) and mouse clique (short). Although controlling the mouse pointer requires only a single touch, we implemented a dual finger gesture e.g. for scrolling on a webpage.

3 Discussion and Outlook

To gather initial feedback from potential users, we had six experienced computer users experiment with the system's functionality[1] for about fifteen minutes. We observed a noticeable effect of keyboard texture as test subjects controlled the mouse pointer. Standard keyboards are structured with indentations or gaps between the keys through which the user's fingers can navigate. However, the uneven shape of keycaps is not an optimal surface for precisely controlling a mouse pointer. A future solution to make the keyboard smoother may be the use of a foil keyboard.

The current prototype has a width-/height-ratio of 1.5, which is between the typical screen ratios of 4:3 and 16:9. Although this ratio does not fit exactly, none of the users noticed this difference. We note that these findings might change if we apply the sensing capability to a complete keyboard thus generating a larger mismatch. Future work will focus on the question of whether it makes more sense to use a complete keyboard as an absolute pointing device or just its separate number block.

Since GF's sensor surface can be used to detect both pointer movements and button strokes, the technology could be used in Ultra-Mobile Personal Computers (UMPC), which have inherently compact designs. Combining the keyboard and the touchpad in such a way that they use the same surface is worthwhile in such devices.

References

1. Rekimoto, J., Ishizawa, T., Schwesig, C., Oba, H.: PreSense: interaction techniques for finger sensing input devices. UIST 2003, pp. 203-212. ACM, New York, NY (2003)
2. Rekimoto, J., Oba, H., Ishizawa, T.: SmartPad: a finger-sensing keypad for mobile interaction. Proc.: CHI 2003, 850-851. ACM, NY (2003)
3. Rosenberg, I. D., Grau, A., Hendee, C., Awad, N., and Perlin, K.: IMPAD: an inexpensive multi-touchpressure acquisition device. Proc. CHI EA'09. ACM, NY, 3217-3222 (2009)
4. Rekimoto, J.: ThumbSense: Automatic input mode sensing for touchpad-based interactions. In: CHI 2003 Late braking (2003)
5. Tsukada, Y., Hoshino, T.: Layered touch panel: the input device with two touch panel layers. Proc. CHI 2002 Interactive Poster, 71-80 (2002)
6. Tsukuda, Y., Hoshino, T.: PointingKeyboard: An input device for typing and pointing. Proc. Workshop on Interactive Systems and Software (WISS 2002) (in Japanese) (2002)
7. Fallot-Burghardt, W., Fjeld, M., Speirs, C., Ziegenspeck, S., Krueger, H., Läubli, T.: Touch&Type: a novel pointing device for notebook computers. Proc.: NordiCHI 2006, vol. 189, pp. 465-468. ACM, NY (2006)
8. Baxter, L. K.: Capacitive Sensors, Design and Applications. IEEE Press, NY (1997), http://www.capsense.com/capsense-wp.pdf; accessed 19.3.2009
9. Mackey, B. L., Golovchenko, M.: Capacitive Sensing Apparatus Designs, patent (2007)
10. Hinckley, K., Sinclair, M.: Touch-sensing input devices. Proc. CHI 1999, 223-230 (1999)
11. Fallot-Burghardt, W., Speirs, C., Ziegenspeck, C., Krueger, H., Läubli, T.: Touch&Type™: a Novel Input Method for Portable Computers. Proc. INTERACT2003, pp. 954-957 (2003)
12. Rekimoto, J.: SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. Proceeding of the SIGCHI conference on Human factors in computing systems: Changing our World, changing ourselves 2002, pp. 113-120
13. Analog Devices Data Sheet AD 7147, http://www.analog.com/static/imported-files/Data_sheets/AD7147.pdf; accessed 22.05.2009