國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

# 結合鍵盤與觸控板之使用者輸入介面設計

# FlickBoard: Enabling Trackpad Interaction with Automatic Mode Switching on a Capacitive-sensing Keyboard

鄭達陽

Ta-Yang Cheng

指導教授：陳彥仰博士

Advisor: Mike Y. Chen, Ph.D.

中華民國 103 年 11 月

November, 2014

國立臺灣大學
敦品勵學・愛國愛人

# 誌謝

感謝實驗室的各位夥伴鼎力相助，以及團隊裡的夥伴們的通力合作。沒有你們就沒有這些成果。感謝我的指導教授陳彥仰老師，在研究的過程中給予了許多的建議以及啟發。感謝余能豪教授在整個計畫的進行過程中在器材和知識上的協助。最後感謝所有在過程中幫助我的朋友們，沒有你們就沒有這份研究。

# 摘要

本篇論文描述一個將傳統鍵盤與操作指標裝置的觸控板合併在同一個互動空間中的互動輸入系統，FlickBoard。本系統透過將兩個經常交互使用的傳統輸入裝置合併成一個裝置來降低使用者的手部移動距離，並且可以全自動的切換輸入模式。本團隊開發的原型機種將一個58x20的電容式觸控線路嵌入在矽膠鍵盤膜上，並且直接固定在一般的孤島式鍵盤上。本系統利用機器學習的演算法來分辨使用者目前的意圖，並且自動啟用鍵盤或者觸控板。目前本系統在5次交叉驗證中可以達到98%的準確率。

國立臺灣大學
敦品勵學・愛國愛人

# Abstract

We present FlickBoard, which combines a touchpad and a keyboard into the same interaction area to reduce hand movement between separate keyboards and touchpads. It supports automatic input mode detection and switching (ie. touchpad vs keyboard mode) without explicit user input. We developed a prototype by embedding a 58x20 capacitive sensing grid into a soft keyboard cover, and uses machine learning to distinguish between moving a cursor (touchpad mode) and entering text (keyboard mode). We conducted experimental studies that show automatic mode switching classification accuracies of 98% are achievable with our technology. Finally, our prototype has a thin profile and can be placed over existing keyboards.

# Contents

國立臺灣大學 · 博碩士論文 ·

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Operating GUI system requires both pointing devices and text input devices. However, most of the commercially available computers put these two devices at two adjacent positions, which requires hand repositioning while switching between two devices. Past works[1, 3, 6] tried to solve this problem by enabling touch sensing capability on the keyboard. In this work we take a step further to make the sensing layer that is easy to be added on the off-the-shelf keyboards. Furthermore, the key issue of the dual functional keyboard is how to automatically switch the pointing mode to the typing mode and vice versa. To our knowledge, this issue has not been solved yet.

We present FlickBoard, a keyboard cover with capacitive touch sensing film to collect user's usage data and design an automatic mode switching algorithm based on users' intention.

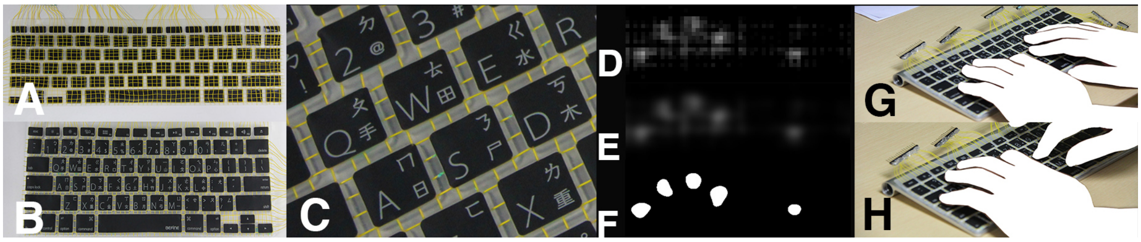

Figure 1.1: We present a keyboard cover with capacitive touch sensing capability which automatically disables itself while typing. Sensing wires are embedded into a typical keyboard cover (A), the modified cover is then put on an off-the-shelf keyboard (B). The sensing grid is all over the keyboard with 0.5cm grid size (C). This results in a low-resolution raw intensity image when hands are near the surface of the keyboard (D). The image is then processed to obtain touched areas (E+F). The raw image can also be used to robustly recognize whether user wants to type on the keyboard (G) or to control cursor with touchpad (H) using a machine learning-based classifier.

# Chapter 2

# Related Work

## 2.1 Co-located Keyboard and Touchpad

Previous research has shown that co-locating two devices together will improve user performance[1]. However, the integrated device still requires manual mode switching to avoid false triggering of pointing device. ThumbSense[5] tried to implement an automatic input mode switching for keyboard by maintaining a state machine controlled by touchpad and keyboard event. Although it helps users keep their fingers on the home row, it still requires users to move their thumb onto the touchpad. Longpad[2] has shown that a larger touchpad occupies the whole area below keyboard can enable more possibilities for interactions.

## 2.2 Keyboard with Motion Gesture

Type–Hover–Swipe[6] implemented a modified keyboard with infra-red proximity sensors that recognizes in-air hand gestures and obtains coarse finger position. Some new interaction techniques for in-air and on-surface gesture on keyboard are explored. The depth map generated by infrared range finder is fast and stable, but the finger position obtained by the system is too rough to control mouse cursor because the sensors were interspersed between the key caps. Although the keyboard supports rich motion gestures, it still cannot precisely track finger movement on keyboard surface. Capacitive sensing technology, in contrast, can obtain higher resolution image under this condition.[3]

# Chapter 3

# System Overview

Our system consists of four parts: 1) sensing film, 2) capacitance-to-digital converters, 3) graphical recognition system and 4) automatic mode switching predictor.

## 3.1 Sensing Cover

We built a capacitive sensing grid on a commercially available silicone keyboard cover. The modified cover was placed over an Apple wireless keyboard. We connect ground end to the body of Apple keyboard to stabilize the readings. The grid consists of 58 vertical and 20 horizontal 30 AWG cooper wires. Each alphabetic and numeric key has 4x4 wires under them. (Figure 3.4) With mutual capacitance sensing technique, each cross point of vertical and horizontal wires can be a single sensing point, so the film can capture a 58x20 frame. The sensing resolution could be higher if the conductive pattern is directly printed on the cover with higher line density. With this modified keyboard cover, we can enable touch sensing capability on by simply putting it on an unmodified keyboard.

## 3.2 Capacitance To Digital Converters(CDC)

To measure the change of mutual capacitance value of the sensor grid cross points, we referenced the design of SmartSkin [4] and built a customized CDC.

The main idea of this design is to measure the signal reduction of the square wave

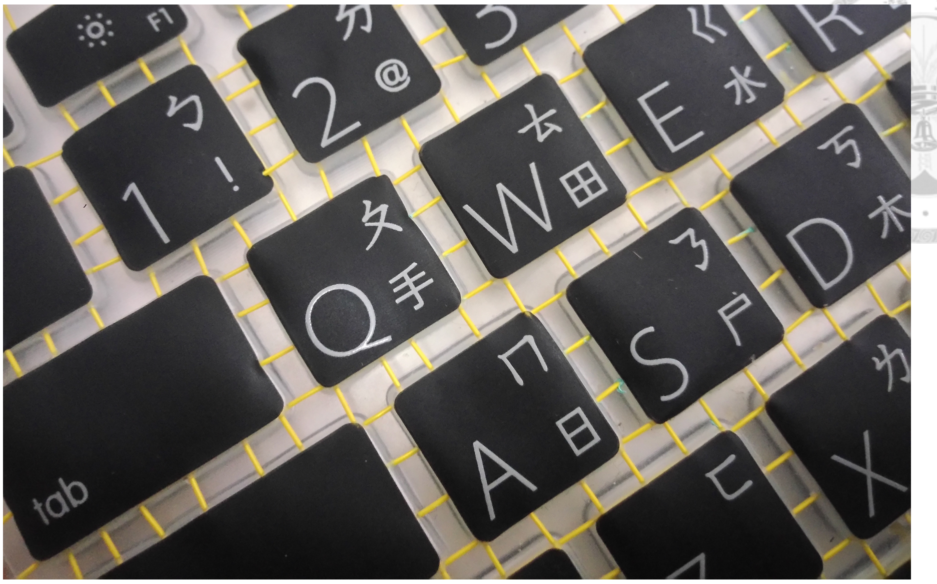

Figure 3.1: Keyboard film with cooper wire embedded. Each standard alphabetic and numeric key has 4x4 wire under them.

signal passed through the sensor film, which can be viewed as a very small capacitor. The square wave signal generated by a programmable clock generator is passed into sensor films through analog demultiplexers, so we can raster scan through all the 58 vertical wires by switching between the channels. 20 OP-Amps are connected to the horizontal wires of the sensor grid, amplifying the weakened signal by a factor of 5 for further processing. We remove the noise generated by circuits nearby with a simple lock-in amplifier, the lock-in amplifier takes noisy signals as input and outputs the signal strength of the target frequency. The CDC also has a analog subtractor for hardware-based background substraction. The CDC samples the output level of analog subtractor with a 10-bit Analog to Digital Converter(ADC) and send the data back to a computer with standard USB serial device. The whole capacitive sensing system can be designed to be smaller and portable.

## 3.3 Sensor Characteristics and Data

The CDC currently is capable of raster scanning through the sensor grid at 13 Hz. The CDC subtracts background image internally. Calibration of the sensor is done by sampling through the sensor grid for 10 times. The data generated by CDC can form a 58x20 pixels

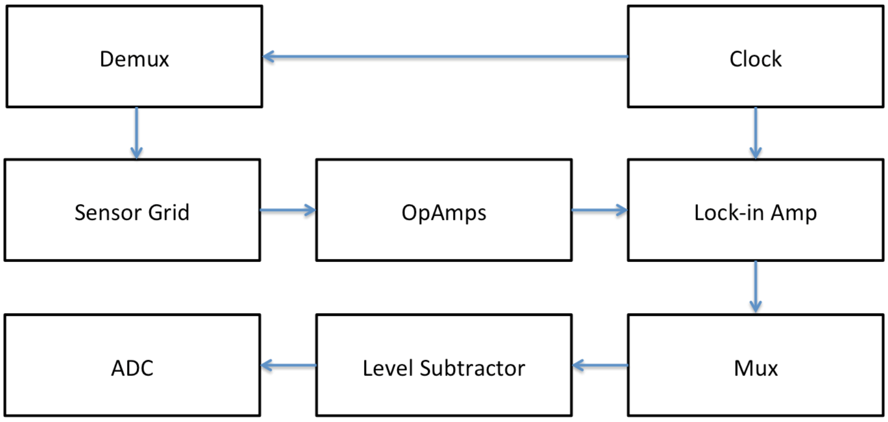

Figure 3.2: CDC circuit diagram. The switch and RC low-pass filter are the main components of the lock-in amplifier.

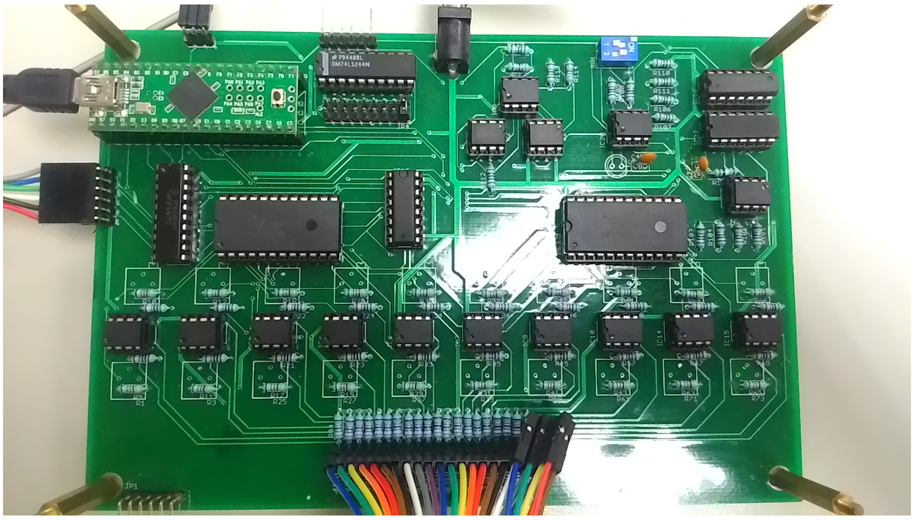

Figure 3.3: The final version of CDC printed circuit board.

intensity, each pixel has 10-bit resolution. The sensor grid only responds to conductive object in a very short range(<0.2cm).

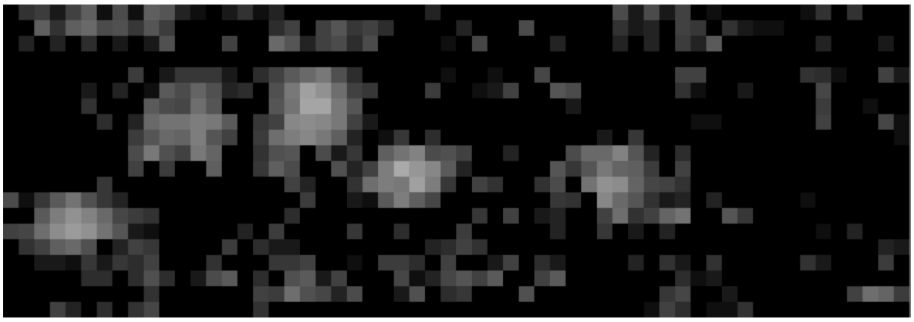

Figure 3.4: Raw image of the obtained capacitive sensor data.

## 3.4 Graphical Recognition System

The 58x20 10-bit intensity image is scaled up to 464x160 10-bit image with nearest-neighbor interpolation to provide more accurate cursor positioning capability. (Figure 3.4) Gaussian filter is then applied to the image for smoother blob images. Each row of the filtered image is subtracted with the mean of the row since we found that the sensor value will be interfered when there are some other touch points on the same horizontal sensing wire.(Figure 1.1.E) The image is binarized with a simple local adaptive thresholding algorithm. Finally, the system detects blobs in the binarized image as touched points. (Figure 3.5) The calculated blob positions are filtered with a Kalman filter to stabilize blob position and make cursor controlling possible.

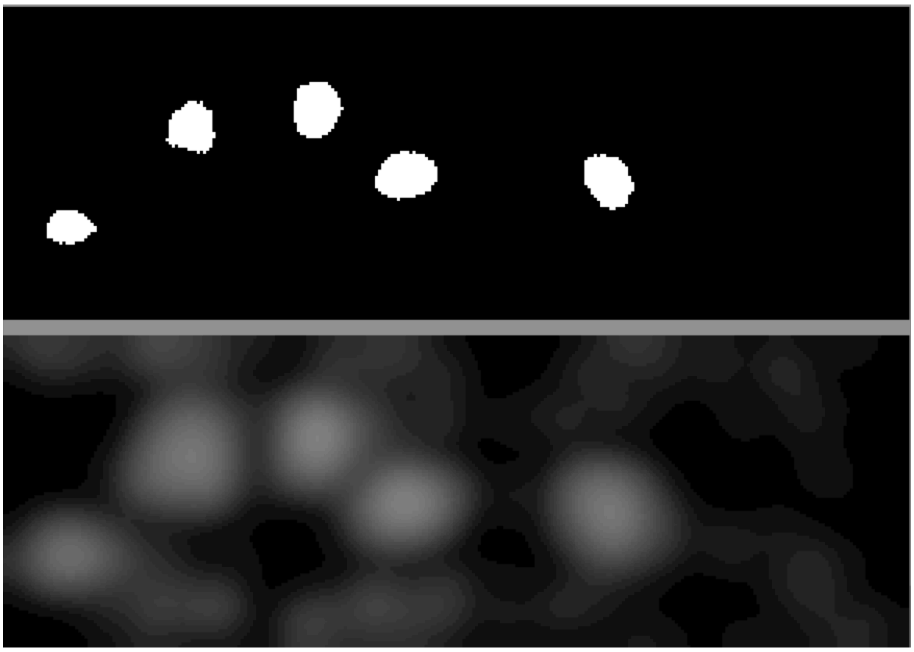

Figure 3.5: Blurred (lower one) and binarized (upper one) images.

## 3.5 Automatic Mode Switching Prediction

We also implemented Motion Signature[6] to recognize whether user is trying to use pointing device or not. Since the sample rate is relatively lower (13Hz) compared to the original condition(325Hz), we reduced the referenced frames to only 10 frames in the process of building motion history image(MHI). Also, we removed binary-MHI(bMHI) from the original MHI implementation because intensity-MHI already provides enough accuracy for recognizing user intention. We classify the calculated MHIs with Random Decision Forests(RDF), the same classifier used in Type–Hover–Swipe[6].

國立臺灣大學 · 敦品勵學 · 愛國愛人 ·

# Chapter 4

# Training Data

In order to train a classifier to recognize the current operation performed by users, we need to collect some usage data with ground truth.

We recruited 30 participants (15 females, 15 males, mean age 21) with an on-line form. All participants are right-handed. In the training session, participants were asked to use FlickBoard to fill in a questionnaire. In the testing session, participants are asked to use FlickBoard to perform the following tasks: 1) Type a specified sentence in the text processing software. 2) Change the font size of the sentence by moving mouse cursor to select different size on the menu bar. 3) Continue tying another sentence and their own name. 4) Insert a picture in the document with menu button, and resize the picture with mouse cursor. 5) Close the text processing software and open a web browser. 6) Type ẅww.facebook.comïn the location bar. 7) Browse the social network site, comment on one of a post on the news feed. 8) Scroll back to the top of the page and upload a photo. 9) Add some comment on the uploaded photo. Participants were not allowed to use hot keys. They can only type and controll the cursor or use scrolling gesture.

We designed the tasks to meet the following criteria: 1) The tasks must be using both keyboard and touchpad alternately. Switching between two devices occurs frequently. 2) The task must be simple enough for the users to perform while they also need to operating an extra foot pedal. 3) The overall usage time of keyboard and touchpad should be as close as possible. This would result in a more balanced dataset, which is better for building a classifier.

The tasks we asked the participants to perform includes document editing and typesetting in a text processing software, and browsing the social network site. All of these tasks were very common tasks for a modern PC user, and should be performed without obstacle. The ground truth of whether the user is trying to use keyboard or touchpad is collected with a foot pedal switch operated by the participant. This foot pedal switch also act as a manual mode switch in the system during the data collection session. Mouse button is triggered with space bar in touchpad mode, and scrolling is controlled with the rightmost area of the keyboard. Video of hand posture and its interaction with keyboard are recorded for further analysis. The total operating time is 187.36 minutes, average operating time is 6.24 minutes per user.

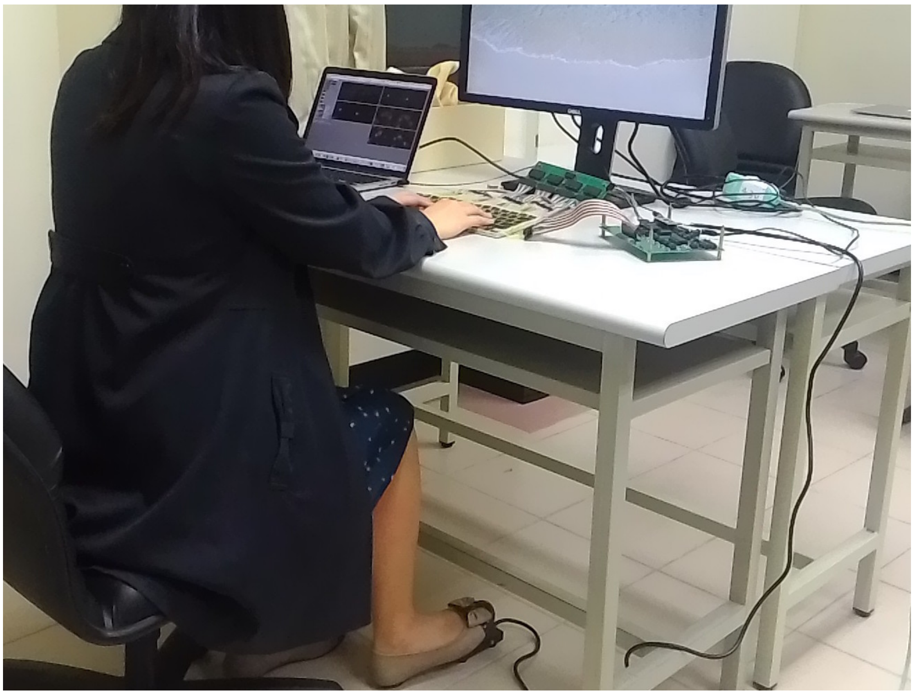

Figure 4.1: Hardware setup of the data collection session. A foot pedal is placed under the table to act as a manual mode switch.

國立臺灣大學 · 敦品勵學 · 愛國愛人

# Chapter 5

# Feature Extraction

We collected 150007 frames from the training data collection session, 25739 were keyboard frames, 61896 were touchpad frames. The remaining 62372 frames were frames without any touched blobs, we call it blank frames. Since those frames don't contain any touched blobs, we can assume no user operation is present at this moment, so we cannot classify it into neither keyboard nor touchpad frames. The blank frames were removed from training data while building a classifier, and directly skipped while running on a real-time interaction system.

We first tried to build a classifier with MHIs generated with filtered 58x20 10-bit image, the recognition rate and result were both quite impressing. But the classifiers trained with higher resolution images were more sensitive with touched position, which requires too much training data, a 30-minute single user training data cannot generate a usable classifier.

We then decided to build another classifier with down-scaled 29x10 10-bit image, it significantly reduced the amount of data required to build a working classifier.

# Chapter 6

# System Evaluation

We evaluate our system by running various cross-validation tests with usage data collected with 30 participants. In this section, we describe the details of the validation process and result.

## 6.1 Testing Parameters

The forest size improves the performance of RDF classifier at a linear cost of time, to build a realtime interaction system, we set the number of trees to 30 for more accurate recognition.

While running the experiments, we found the number of frames referenced while building Motion Signature ($N_f$) can strongly affects the performance on recognizing user intention, do we need to find out a good parameter. We run a 5-fold cross validation with 1-20 frames referenced, the results are shown at Figure 6.1. We can find that the recognition rate is very low when $N_f$ is 1, then gradually rising while $N_f$ is increasing. The recognition rate stabilized around 98% when $N_f = 30$. This means we only need to remember 30 frames to achieve a good recognition rate, and we can recognize the user's intention with about 2.3 seconds of previous surface action. According to the result, we set $N_f = 30$ in the following experiments.

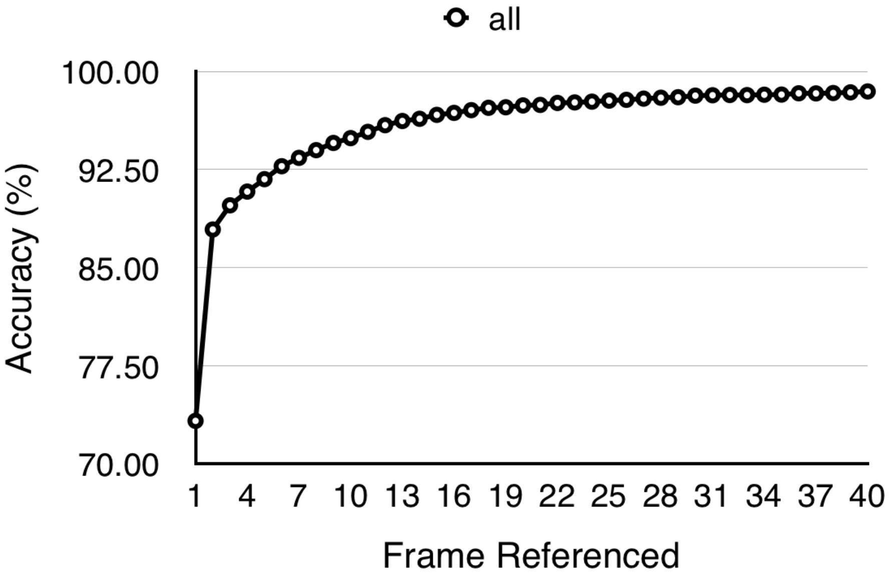

Figure 6.1: Classification accuracy of 5-fold cross validation for each user and leave-one-user-out cross validation. X axis is number of frame referenced while building Motion Signature ($N_f$)

| | Keyboard | Touchpad | Blank |
|---|---|---|---|
| Keyboard | 24709 | 1030 | 0 |
| Touchpad | 595 | 61301 | 0 |
| Blank | 0 | 0 | 62372 |

Table 6.1: Confusion matrix for three states,keyboard state, touchpad state and unknown state in a 5-fold cross validation. Mean accuracy is 98.14%

## 6.2 Performance

We run a 5-fold cross validation with each participant's own data with parameters shown above, the averaged overall recognition rate is 98.83%, with maximum 99.52% and minimum 96.91%. We also conducted a leave-one-user-out cross validation, the averaged overall recognition rate is 83.71%, with maximum 94.62% and minimum 49.53%. The accuracy of leave-one-user-out cross validation strongly depends on users' behavior. If a user's behavior is very similar with another one, his/her accuracy will be relatively higher. When we build a shared classifier with all participant's data, the performance is 98.42% recognition rate in a 5-fold cross validation. The confusion matrix of shared classifier is shown at Table 6.1.

# Chapter 7

# Discussion

The performance evaluation of FlickBoard shows a satisfying result with usage data collected from a variety of participants. To our knowledge, this is the first time that a automatic mode switching keyboard and touchpad hybrid device is presented. Because our implementation is based on RDF, it is very easy to adapt to new user bahavior with small amount of training data (about 6 minutes). In this section, we discuss about the advantages, limitations and future work of this prototype.

## 7.1 Preliminary User Behavior Observation

While we expect most of the users will rest their fingers on the surface of keyboard, we observed that a large part (19 out of 30) of the users lifted their left hand while using touchpad function. None of the participants can explain the reason of this behavior. However, we spotted that when those users were typing, both hands will lift for smoother hand movement. Users might apply their experience of using traditional keyboard onto FlickBoard while controlling cursor, so they lifted their left hand while right hand is performing cursor movement operation.

All of the participants use only one finger to move cursors without any instructions. Participants report that they directly adapted their previous experience of using traditional touchpad, so they use only one finger to touch on the surface of FlickBoard.

Most of the users will try to use touchpad in a very small area, about 3x3 cm. 20 of 30 users operate cursors on the center area of the keyboard, 9 users operate on the center of right side, the other 1 user operates on the bottom of right side.

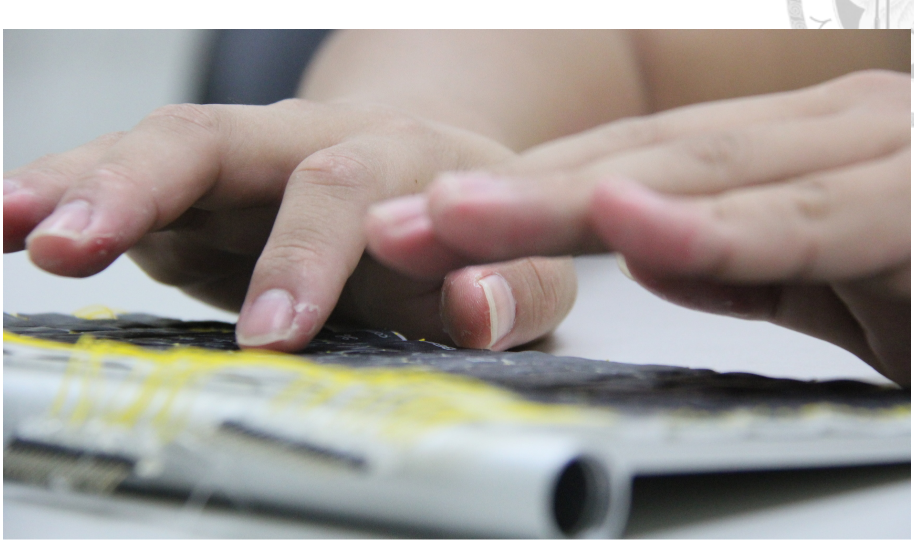

Figure 7.1: A majority of participants raised their left hand and use one finger of right hand to perform cursor movement in the trackpad mode.

## 7.2 Uneven Surface

Many users reports that the surface of FlickBoard is too uneven to perform smoother cursor pointing operation. Users' hands may get stuck between the key caps, which makes user harder to move their finger to desired position. This drawback can be solved with some physical modification, such as adding a mechanical structure to lift case of a Chiclet keyboard to the top of key caps, forming a flat and smooth platform for users to perform surface operation. We further built an automatically lift keyboard with our automatic mode switching predictor. We will conduct an usability testing with this form factor in the future.

## 7.3 Higher Frame Rate and Resolution

Frame rate of current prototype may be too low for some time-critical interactions. There are two major bottleneck of current implementations: sampling speed of the lock-in amplifier and speed of MCU.

Currently, the sampling speed of lock-in amplifier is bounded by two factors: the RC time constant of RC low-pass filter and sampling speed of ADC. Both of them can be improved by using better implementation options, which requires modification of hardware design.

On the other hand, we can save more MCU computing power by dividing the raster scanning process into two stage. We can scan a lower resolution image first, calculate possible touch blobs position with it, and perform a higher resolution raster scanning around touched blobs. This modified process is faster in most of the cases (number of touched blobs $< 5$), and does not sacrifice precision. As a result, the frame rate can be higher without modifying hardware setup.

# Chapter 8

# Conclusions

In this work, we designed a keyboard add-on that is easy to install to enable touch capability on the regular keyboard. Our system can also automatically detect user's intention of switching between pointing and typing with a realtime recognizer. The proposed touch sensing technique has higher resolution over the previous works on performing gestures on keyboard, it can also be used for bimanual multitouch gestures or more.

# Bibliography

[1] W. Fallot-Burghardt, M. Fjeld, C. Speirs, S. Ziegenspeck, H. Krueger, and T. Läubli. Touch&amp;type: A novel pointing device for notebook computers. NordiCHI '06, pages 465–468, New York, NY, USA, 2006. ACM.

[2] J. Gu, S. Heo, J. Han, S. Kim, and G. Lee. Longpad: A touchpad using the entire area below the keyboard of a laptop computer. CHI '13, pages 1421–1430, New York, NY, USA, 2013. ACM.

[3] I. Habib, N. Berggren, E. Rehn, G. Josefsson, A. Kunz, and M. Fjeld. Dgts: Integrated typing and pointing. In T. Gross, J. Gulliksen, P. Kotz√©, L. Oestreicher, P. Palanque, R. Prates, and M. Winckler, editors, *INTERACT 2009*, volume 5727 of *Lecture Notes in Computer Science*, pages 232–235. Springer Berlin Heidelberg, 2009.

[4] J. Rekimoto. Smartskin: An infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 113–120, New York, NY, USA, 2002. ACM.

[5] J. Rekimoto. Thumbsense: Automatic input mode sensing for touchpad-based interactions. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 852–853, New York, NY, USA, 2003. ACM.

[6] S. Taylor, C. Keskin, O. Hilliges, S. Izadi, and J. Helmes. Type-hover-swipe in 96 bytes: A motion sensing mechanical keyboard. CHI '14, pages 1695–1704, New York, NY, USA, 2014. ACM.