Identification Method of Digits for Expanding Touchpad Input

Takuto Nakamura(✉) and Buntarou Shizuki

University of Tsukuba, 1-1-1 Tennoudai, Tsukuba, Ibaraki, Japan
{nakamura,shizuki}@iplab.cs.tsukuba.ac.jp

Abstract. A method is presented for identifying the digits, i.e., the thumb and/or finger(s), that touch a touchpad as a means to expand the input vocabulary available on a touchpad. It will enable application designers to assign different commands to touch gestures performed with the same number of digits and the same movement but with different digits. No additional sensors are required for identification; instead the digits are identified on the basis of machine learning using only data acquired from a mutual-capacitance touchpad as learning data. Experimental results revealed an average identification accuracy of 86.3%.

Keywords: Digit identification · Touch gestures · Machine learning

1 Introduction

The touchpad has become the most common pointing device for laptop computers. The system executes a command in accordance with the number of digits (i.e., the thumb and/or finger(s)) that touch on the touchpad and their movement. However, due to the limited input vocabulary available on a touchpad, an application that requires frequent switches between many tools and modes forces users to frequently select graphical user interface (GUI) elements such as items in tool menus and buttons in menu bars or to use keyboard shortcuts. For example, in order to draw or select figures when using a graphical editor, the user first selects the desired tool by either choosing it from a toolbar or using the keyboard shortcut assigned to the tool and then touching the touchpad to manipulate the tool. For such applications, it would be useful to be able to switch tools or modes at the same time as making a touch gesture.

Our goal is to expand the input vocabulary available on a touchpad by developing a method for identifying the digits that touch the touchpad; this identification will enable application designers to assign different commands to touch gestures performed with the same number of digits and the same movement but with different digits (Fig. 1). In this paper, as a first step towards achieving this goal, we present a method for identifying the digits, i.e., the thumb and/or finger(s), that touch the touchpad. No additional sensors are needed for identification; instead the digits are identified on the basis of machine learning using only data acquired from a mutual-capacitance touchpad as learning data.

M. Kurosu (Ed.): HCII 2020, LNCS 12182, pp. 463–474, 2020.
https://doi.org/10.1007/978-3-030-49062-1_31

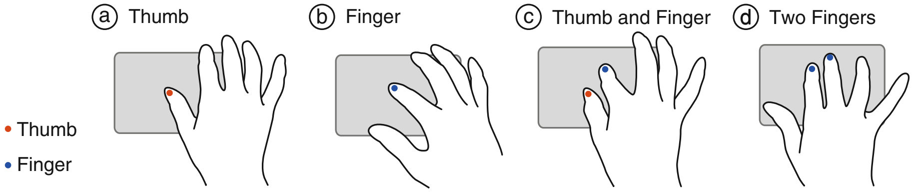

Fig. 1. Touchpad input based on thumb-or-finger identification. Touch gestures using (a) one thumb or (b) one finger can be assigned to different commands; similarly, touch gestures using a combination of (c) a thumb and a finger or (d) two fingers can be assigned to different commands.

2 Related Work

Many previous studies similar to ours have focused on expanding touchpad input by identifying the digits used.

2.1 Expanding Touchpad Input

Cui et al. [4] developed a method that places a virtual keyboard on the touchpad area while the user is pressing a specific modifier key enabling the user to execute a shortcut command by tracing the keys in the order of the spelling of the command name. In the method developed by Berthellemy et al. [2], the touchpad is divided into a grid of areas and a command is assigned to each area; the user can quickly execute a command by pressing a dedicated key on the keyboard and then tapping the corresponding area on the touchpad. They also proposed displaying command menus at the four vertices and on the four sides of the screen, enabling the user to execute a command by swiping the corresponding touchpad bezel. Fruchard et al. [5] developed a method in which the touchpad area is divided into a checkerboard pattern, and a stroke gesture from one area to another is used to trigger a specific command; the stroke pattern and command name are displayed on the screen at execution for visual assistance. Ikematsu et al. [10] added button-like and pointing stick-like inputs to a touchpad by attaching pressure-sensitive sensors to the area next to the touchpad and pasting the electrodes to the touchpad surface; the pressure exerted on a sensor was measured and taken as the leakage current from the corresponding touch surface electrodes.

Several methods expand touchpad input by using touchpads with special structures. Jung et al. [11] made it possible to give the user tactile feedback and feedforward in a form recognizable with the fingers by using a touchpad implemented using a 40×25 array of vertical-type Braille display module pins. Gu et al. [7] extended the width of the touchpad to that of the laptop's keyboard, thereby facilitating the input of long horizontal gestures.

Furthermore, several methods using an infrared proximity sensor array or a capacitive grid enable touchpad input on a keyboard or hover-gesture input on a touchpad [3,9,18,19].

In contrast, our method focuses on expanding touchpad input rather than keyboard input. Additionally, our method can be applied to existing touchpads since it requires no additional sensor or attachment.

2.2 Digit Identification

Digit identification using a capacitive image of a touchscreen to expand input vocabulary on a touchscreen has been explored. Le et al. [12] developed a method that acquires a capacitive image from a smartphone's touchscreen and identifies the digit touching the screen using a convolutional neural network. Experiments showed that its identification accuracy for the thumb (left or right) is 92%. Gil et al. [6] developed a method that identifies the thumb, index finger, and middle finger by acquiring a capacitive image from the touchscreen of a smartwatch and using machine learning (Random Forest). They reported that the identification accuracies during tapping and swiping tasks were 93% and 98% in natural poses and exaggerated poses, respectively.

There are also methods for identifying digits by attaching sensors to the fingers. Gupta et al. [8] developed one that identifies the touch of the index finger and middle finger on the touchscreen of a smartwatch by using infrared proximity sensors attached to the finger pads. Park and Lee [16] developed one that identifies the touch of the index finger, middle finger, and ring finger on the touchscreen of a smartwatch by using a ring with a built-in magnet. The magnetic field is acquired using the smartwatch's built-in magnetic sensor and the fingers are identified using machine learning (Support Vector Machine). With the method of Masson et al. [14], a vibration sensor is attached to each digit (from thumb to little finger), and the digit that detects the largest vibration when there is a keyboard or touchpad input event is determined to be the digit that was used for the input. Benko et al. [1] attached an electromyography sensor to the arm to detect electromyograms that are generated when a user inputs a touch gesture to the touch screen. The fingers used for the gesture are identified by machine learning. The method of Vega and Fuks [20] identifies fingers touching a flat surface with a tag reader by using artificial fingernails with a built-in electronic tag. Furthermore, several reported methods identify fingers by computer vision [13,15,17,21].

In contrast, we focus on digit identification for a touchpad rather than a touchscreen. Moreover, our method requires no additional sensors to identify digits.

3 Identification Method

We focused on the fact that when a digit touches a touchpad, the side of the thumb touches, whereas the pad of a finger touches (see Fig. 2). This means that the shape of the contact surface of the thumb and that of a finger differ. Using this difference, our method identifies the digits touching the touchpad by acquiring data on the contact surfaces from the touchpad and classifying them using machine learning. Our method comprises two phases: the learning phase and the identification phase.

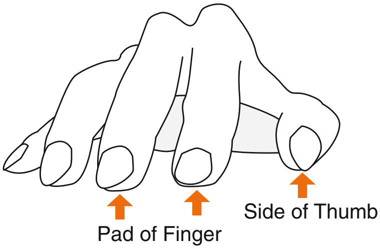

Fig. 2. Contact surfaces of digits touching touchpad.

3.1 Learning Phase

In the learning phase, the system collects data from the touchpad as the user slides each digit of both hands on the touchpad. The system regards the contact area between a digit and the touchpad as an ellipse and uses eight values as data for machine learning: the x and y coordinates of its center, the lengths of the major and minor axes, the angle of the major axis, the area, the ellipticity, and the contact surface density. The system collects these values (except the ellipticity) from the touchpad and calculates the ellipticity using the collected values. To enable collection of learning data covering the entire touchpad area, we divided the touchpad area into $5 \times 5 = 25$ areas and collected learning data from 30 frames for each area. In total, the system collects data from 15,000 frames (5 digits $\times$ 2 hands $\times$ 2 directions $\times$ 25 areas $\times$ 30 frames) for each user.

Next, the system generates a machine learning model from the collected data. For simplicity, here we use a value indicating thumb or finger (i.e., two labels) as the objective variable; we use the learning data (i.e., the data from the 15,000 frames) as the explanatory variables.

3.2 Identification Phase

The system, in real-time, identifies the digits at the moment that touching begins. That is, every time the system acquires data on the contact surface of a digit touching the touchpad, it uses the machine learning model generated in the learning phase to identify the digit. If two or more digits touch the touchpad, the system applies this process to each contact surface. The predictor variables are the same as the explanatory variables in the learning phase (i.e., eight values). The response variable is the result of digit identification (thumb or finger).

3.3 Implementation

We implemented our method in a system as a macOS application. We used Core ML and Create ML as machine learning frameworks. As a model for machine learning, we used Gradient Boosting, which was selected by Create ML as the optimal machine learning algorithm among five algorithms: Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine, and Logistic Regression. We used MultitouchSupport for acquiring raw data from the touchpad.

4 Experiment

We conducted an experiment to investigate the identification accuracy of our method.

4.1 Apparatus and Participants

We used a laptop computer (Apple, MackBook Pro 13-inch, 2017) with a built-in touchpad (size: 134 mm × 83 mm, resolution: 26 × 18 matrix). We pasted LCD protective film on the surface of the touchpad, on which 5 × 5 = 25 areas where drawn with a maker pen so that the participants could see the 25 areas.

We recruited 12 full-time undergraduate and graduate students from a local institution (all male, mean age = 23.2, 11 right-handed) who usually use a laptop computer with a touchpad.

4.2 Tasks

The participants were asked to perform two tasks: a learning-data sampling task and a test-data sampling task.

Learning-Data Sampling Task. As shown in Fig. 3, the participants slid a digit on the touchpad horizontally and vertically until all 25 areas turned red, indicating that learning data from 30 frames for an area had been collected. We asked the participants to sit on a chair with a laptop computer on a desk in front of them and to slide their digit in both directions horizontally (i.e., left to right and right to left). We collected touch data from 30 frames for each digit, area, and direction in a fixed order (digit types: thumb → little finger, hands: right → left, directions: horizontal → vertical). We instructed the participants not to press the touchpad.

Test-Data Sampling Task. An image of the application used in this task is shown in Fig. 4. In each session, the participant was asked to touch each area once with each digit. The order of digit types was presented randomly by the application. For every digit type, the order of the area to be touchpad was presented randomly. The next area was not presented until the correct area was touchpad.

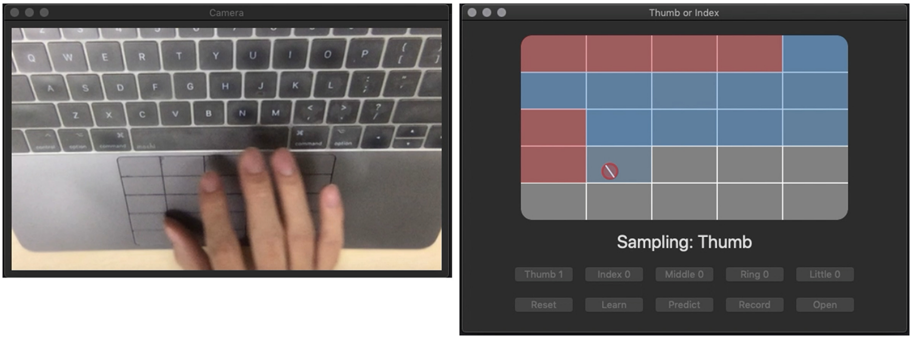

Fig. 3. Image of a participant sliding right thumb horizontally in learning-data sampling task, and image showing collection status of each area (red: completed, blue: partially completed, gray: not started). (Color figure online)

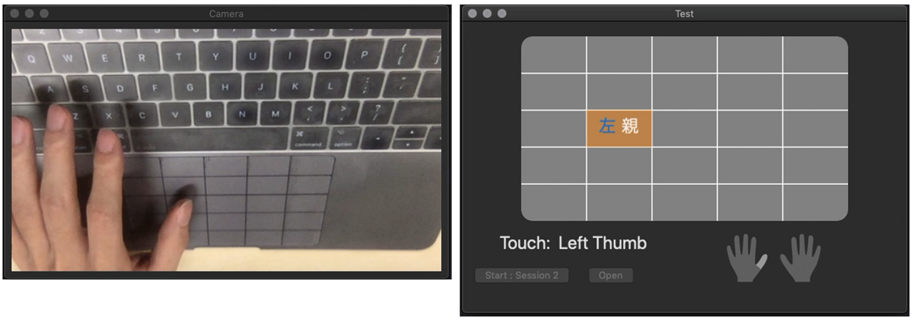

Fig. 4. Image of participant touching an area with his left thumb in the test-data sampling task.

To examine the robustness of our method, we collected test data under three conditions: *Desk Condition*, *Lap Condition*, and *Bed Condition*, as shown in Fig. 5, as shown in Fig. 5. In Desk Condition, the participant sat on a chair with a laptop computer on a desk in front of him. In Lap Condition, he sat on a chair with a laptop computer on his lap. In Bed Condition, he lay on a foldable camp bed (AZUMAYA, LFS-709GY, 190 cm $\times$ 67 cm $\times$ 35 cm) and faced downwards toward the laptop computer.

4.3 Procedure

After listening to a brief description of the purpose of this experiment, the participants gave their informed written consent. They then carried out the learning-data sampling task and then six sessions of the test-data sampling task. The order of conditions for each participant is shown in Table 1. They were forced to take a break of two minutes or more between sessions in the test-data sampling task.

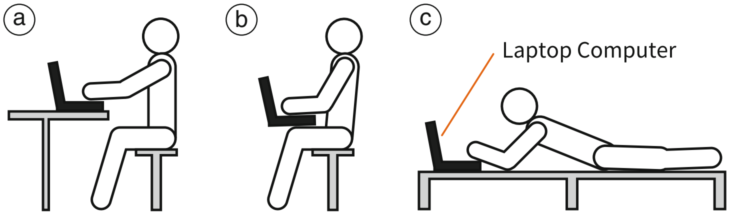

Fig. 5. Conditions for data collection in test-data sampling task: (a) *Desk Condition*, (b) *Lap Condition*, and (c) *Bed Condition*.

Table 1. The order of conditions of the test-data sampling task.

Participant	Sessions 1, 4	Sessions 2, 5	Sessions 3, 6
P_1, P_2	*Desk*	*Lap*	*Bed*
P_3, P_4	*Desk*	*Bed*	*Lap*
P_5, P_6	*Lap*	*Desk*	*Bed*
P_7, P_8	*Lap*	*Bed*	*Desk*
P_9, P_{10}	*Bed*	*Desk*	*Lap*
P_{11}, P_{12}	*Bed*	*Lap*	*Desk*

After finishing both tasks, the participants were given 15.75 USD for their participation. The experiment took approximately 70 min per participant.

4.4 Result

We trained a machine learning model using the data collected in the learning-data sampling task and investigated the accuracy of thumb-or-finger identification (*TF accuracy*) under every condition by using the data collected in the test-data sampling task as test data. As shown in Fig. 6 left, the average identification accuracy for all conditions was 86.3% (SD = 3.41%). Among the three conditions, the Desk Condition had the highest accuracy (87.9%, SD = 5.38%). However, a one-way ANOVA did not show a significant difference between the three conditions ($F_{2,33} = 1.1888$, $p = 0.3173 > 0.05$).

We also investigated TF accuracy when the hand used for operation was distinguished (*TFLR accuracy*) for every condition. As shown in Fig. 6 right, the average identification accuracy for all conditions was 58.3% (SD = 7.63%). Similar to TF accuracy for which the hand used for touching was not distinguished, there was no significant difference between the three conditions ($F_{2,33} = 0.1216$, $p = 0.8859 > 0.05$) with a one-way ANOVA. Additionally, a one-way ANOVA showed a significant difference between TF accuracy and TFLR accuracy ($F_{1,22} = 123.11$, $p = 1.763 \times 10^{-10} < 0.01$).

Figure 7 shows the TF accuracy for each area of the touchpad without the left-handed participant (P_6). A one-way ANOVA showed a significant main effect of row ($F_{4,20} = 16.859$, $p = 3.417 \times 10^{-6} < 0.01$). Tukey's honestly significant difference (HSD) pair wise comparison showed significant differences between rows 1 and 2 ($p = 0.0178991 < 0.05$), between rows 1 and 3 ($p = 1.824 \times 10^{-4} < 0.01$), between rows 1 and 4 ($p = 9.1 \times 10^{-6} < 0.01$), and between rows 1 and 5 ($p = 1.08 \times 10^{-5} < 0.01$). This indicates that the bottom edge area of the touchpad has low identification accuracy. In contrast, a one-way ANOVA did not show a significant difference between the five columns of the touchpad.

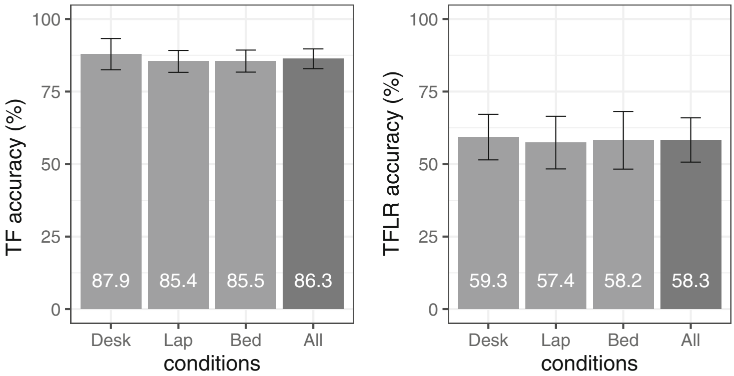

Fig. 6. Accuracy of thumb-or-finger identification (*TF accuracy*) for every condition (left) and when the hand used for operation was distinguished (*TFLR accuracy*) for every condition (right). Error bars represent one standard deviation from the mean.

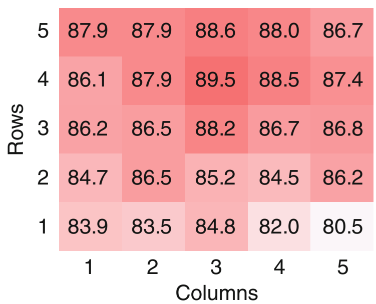

Fig. 7. Identification accuracy for each area of the touchpad (%).

5 Discussion

The average TF accuracy of 86.3% demonstrates the potential of our method for identifying the digit used for touchpad input, whereas the average TFLR accuracy of 58.3% demonstrates its weakness in identifying both the hand and digit used for touchpad input.

Despite the relatively high TF accuracy, further improvement is needed. One strategy for improving TF accuracy is to increase the amount of learning data collected in the learning phase. In our experiment, the participants carried out the learning phase only under Desk Condition. It would be more effective to carry out the learning phase under various conditions for robustness. Furthermore, accuracy could be improved by improving the accuracy of the data acquired from the touchpad. For example, the angle of the major axis acquired from MultitouchSupport becomes unstable or becomes a fixed value ($\pi/2$) when the area of the contract surface is below a certain size.

Moreover, we observed that the accuracy for participants with a shorter thumb was lower than that of those with a longer thumb. We attribute this to the shape of the contact surface of a short thumb being similar to that of the fingers since as short thumb would tend to touch a touchpad by its tip rather than its side. Therefore, it is necessary to examine the effects of various factors, including thumb length, on identification accuracy.

Another strategy for improving accuracy is to exclude the ring and little fingers from the identification targets because these fingers are rarely used for single-finger or two-finger gestures compared with the other digits. This could be effective since reducing the number of classified categories tends to increase accuracy in machine learning approaches.

6 Example Applications

Digit identification has two applications that would make a touchpad more convenient to use.

First, for a graphical editor, the user could use the eraser tool with their thumb and a brush tool with their finger, as illustrated in Fig. 8a, b. Additionally, the user could resize an object by using their thumb and finger and rotate an object by using two fingers, as illustrated in Fig. 8c, d. With these capabilities, the user could perform editing seamlessly without using keyboard shortcuts or GUI elements.

Second, for a text editor, the user could *right-click* with their thumb and *left-click* with their finger, as illustrated in Fig. 9. Moreover, it would possible to move the cursor more slowly (e.g., at half speed) when using the thumb. As a result, the user could perform both fast pointing and fine pointing easily.

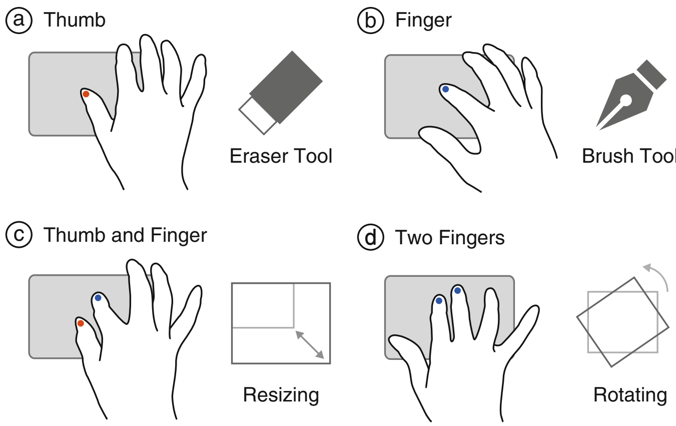

Fig. 8. Example application to graphical editor: (a) using the eraser tool with a thumb, (b) using the brush tool with a finger, (c) resizing an object with thumb and finger, and (d) rotating object with two fingers.

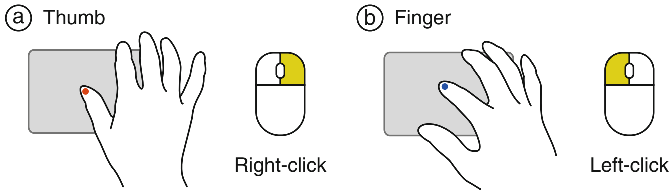

Fig. 9. Example application to text editor: (a) right-clicking using a thumb and (b) left-clicking using a finger.

7 Conclusion

Our method for identifying the digits used to touch a touchpad expands the input vocabulary available on a touchpad. With this method, an application designer can assign different commands to touch gestures performed with the same number of digits and the same movement but with different digits. It is implemented using machine learning. Experimental results demonstrated an average identification accuracy of 86.3%. Future work includes testing the effect of the strategies described in the Discussion on identification accuracy.

References

1. Benko, H., Saponas, T.S., Morris, D., Tan, D.: Enhancing input on and above the Interactive Surface with Muscle Sensing. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS 2009, pp. 93–100. ACM, New York (2009). https://doi.org/10.1145/1731903.1731924

2. Berthellemy, M., Cayez, E., Ajem, M., Bailly, G., Malacria, S., Lecolinet, E.: SpotPad, LociPad, ChordPad and InOutPad: investigating gesture-based input on touchpad. In: Proceedings of the 27th Conference on L'Interaction Homme-Machine, IHM 2015, pp. 4:1–4:8. ACM, New York (2015). https://doi.org/10.1145/2820619.2820623
3. Choi, S., Han, J., Kim, S., Heo, S., Lee, G.: ThickPad: a hover-tracking touchpad for a laptop. In: Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology, UIST 2011, Adjunct, pp. 15–16. ACM, New York (2011). https://doi.org/10.1145/2046396.2046405
4. Cui, W., Zheng, J., Lewis, B., Vogel, D., Bi, X.: HotStrokes: word-gesture shortcuts on a trackpad. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, pp. 165:1–165:13. ACM, New York (2019). https://doi.org/10.1145/3290605.3300395
5. Fruchard, B., Lecolinet, E., Chapuis, O.: MarkPad: augmenting touchpads for command selection. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI 2017, pp. 5630–5642. ACM, New York (2017). https://doi.org/10.1145/3025453.3025486
6. Gil, H., Lee, D., Im, S., Oakley, I.: TriTap: identifying finger touches on smartwatches. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI 2017, pp. 3879–3890. ACM, New York (2017). https://doi.org/10.1145/3025453.3025561
7. Gu, J., Heo, S., Han, J., Kim, S., Lee, G.: LongPad: a touchpad using the entire area below the keyboard of a laptop computer. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 1421–1430. ACM, New York (2013). https://doi.org/10.1145/2470654.2466188
8. Gupta, A., Balakrishnan, R.: DualKey: miniature screen text entry via finger identification. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI 2016, pp. 59–70. ACM, New York (2016). https://doi.org/10.1145/2858036.2858052
9. Heo, S., Han, J., Lee, G.: Designing rich touch interaction through proximity and 2.5D force sensing touchpad. In: Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, OzCHI 2013, pp. 401–404. ACM, New York (2013). https://doi.org/10.1145/2541016.2541057
10. Ikematsu, K., Fukumoto, M., Siio, I.: Ohmic-Sticker: force-to-motion type input device for capacitive touch surface. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI EA 2019, pp. LBW0223:1-LBW0223:6. ACM, New York (2019). https://doi.org/10.1145/3290607.3312936
11. Jung, J., Youn, E., Lee, G.: PinPad: touchpad interaction with fast and high-resolution tactile output. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI 2017, pp. 2416–2425. ACM, New York (2017). https://doi.org/10.1145/3025453.3025971
12. Le, H.V., Mayer, S., Henze, N.: Investigating the feasibility of finger identification on capacitive touchscreens using deep learning. In: Proceedings of the 24th International Conference on Intelligent User Interfaces, IUI 2019, pp. 637–649. ACM, New York (2019). https://doi.org/10.1145/3301275.3302295
13. Malik, S., Laszlo, J.: Visual touchpad: a two-handed gestural input device. In: Proceedings of the 6th International Conference on Multimodal Interfaces, ICMI 2004, pp. 289–296. ACM, New York (2004). https://doi.org/10.1145/1027933.1027980

14. Masson, D., Goguey, A., Malacria, S., Casiez, G.: WhichFingers: identifying fingers on touch surfaces and keyboards using vibration sensors. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST 2017, pp. 41–48. ACM, New York (2017). https://doi.org/10.1145/3126594.3126619
15. Nakamura, T., Shizuki, B.: Distinction system of left and right hands placed on a keyboard of laptop computers. In: Proceedings of the 30th Australian Conference on Computer-Human Interaction, OzCHI 2018, pp. 587–589. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3292147.3292228
16. Park, K., Lee, G.: FingMag: finger identification method for smartwatch. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI EA 2019, pp. LBW2216:1–LBW2216:6. ACM, New York (2019). https://doi.org/10.1145/3290607.3312982
17. Suzuki, Y., Misue, K., Tanaka, J.: A potential exploration of finger-specific interaction. ICIC Express Lett. **6**, 3061–3067 (2012)
18. Taylor, S., Keskin, C., Hilliges, O., Izadi, S., Helmes, J.: Type-hover-swipe in 96 bytes: a motion sensing mechanical keyboard. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems, CHI 2014, pp. 1695–1704. ACM, New York (2014). https://doi.org/10.1145/2556288.2557030
19. Tung, Y.C., Cheng, T.Y., Yu, N.H., Wang, C., Chen, M.Y.: FlickBoard: enabling trackpad interaction with automatic mode switching on a capacitive-sensing keyboard. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, pp. 1847–1850. ACM, New York (2015). https://doi.org/10.1145/2702123.2702582
20. Vega, K., Fuks, H.: Beauty tech nails: interactive technology at your fingertips. In: Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction, TEI 2014, pp. 61–64. ACM, New York (2014). https://doi.org/10.1145/2540930.2540961
21. Zheng, J., Vogel, D.: Finger-aware shortcuts. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI 2016, pp. 4274–4285. ACM, New York (2016). https://doi.org/10.1145/2858036.2858355