

Article

Multi-Touch Interaction Generation Device by Spatiotemporally Switching Electrodes [†]

Masahiro Okamoto and Kazuya Murao * 

Graduate School of Information Science and Engineering, Ritsumeikan University, Kyoto 525-8577, Japan; masahiro.okamoto@iis.ise.ritsumei.ac.jp

* Correspondence: murao@cs.ritsumei.ac.jp

[†] This paper is an extended version of our paper published in Okamoto, M., Murao, K. Construction of a Device that Automatically Generates Various Touch Interactions. In Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2019), London, UK, 9–13 September 2019; pp. 336–338.

Abstract: With the spread of devices equipped with touch panels, such as smartphones, tablets, and laptops, the opportunity for users to perform touch interaction has increased. In this paper, we constructed a device that generates multi-touch interactions to realize high-speed, continuous, or hands-free touch input on a touch panel. The proposed device consists of an electrode sheet printed with multiple electrodes using conductive ink and a voltage control board, and generates eight multi-touch interactions: tap, double-tap, long-press, press-and-tap, swipe, pinch-in, pinch-out, and rotation, by changing the capacitance of the touch panel in time and space. In preliminary experiments, we investigated the appropriate electrode size and spacing for generating multi-touch interactions, and then implemented the device. From the evaluation experiments, it was confirmed that the proposed device can generate multi-touch interactions with high accuracy. As a result, tap, press-and-tap, swipe, pinch-in, pinch-out, and rotation can be generated with a success rate of 100%. It was confirmed that all the multi-touch interactions evaluated by the proposed device could be generated with high accuracy and acceptable speed.

Keywords: touch panel; multi-touch interaction generation; human–computer interaction; interface



Citation: Okamoto, M.; Murao, K. Multi-Touch Interaction Generation Device by Spatiotemporally Switching Electrodes. *Electronics* **2021**, *10*, 1475. <https://doi.org/10.3390/electronics10121475>

Academic Editor: George A. Tsihrintzis

Received: 1 May 2021
Accepted: 15 June 2021
Published: 19 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Devices equipped with a capacitive touch panel, such as smartphones, tablets, and laptops have become widespread. The users can intuitively input the intended operations by performing touch interactions on the touch panel. Touch interaction is superior to the keyboard and mouse input in that it enables rich interaction in a small area. Although small screens such as smartwatches limit input and output space, a synergistic system by allowing input and output on the strap of a smartwatch [1] and a new gesture input using touch and blink synchronization [2] have been proposed. Touch interaction includes tap using one finger and multi-touch using multiple fingers. With multi-touch, it is possible to perform complicated input that is difficult with a tap, and the range of input expression expands. Some studies have also proposed new touch interactions by creating their recognition algorithms [3] and by using buttons attached to the edge of the touch panel [4]. Thus, research on input methods for touch panels has been actively conducted. On the other hand, a tangible interface using a magnetic rubber sheet and conductive material [5], and a sticker with a striped conductive part printed on the sheet [6], have been proposed as methods to generate touch input without directly touching the touch panel.

In this paper, we propose a device that dynamically generates multi-touch interactions using a printable circuit and a microcontroller. The proposed device generates multi-touch interactions such as swiping and pinching with one interface without installing any new software on the touch panel. If there is a device that generates multi-touch interactions

on behalf of the user, high-speed, continuous, and hands-free touch input can be realized, and it can be expected to improve work efficiency and automation using touch panels. Integrating the proposed device into a tangible device or stylus, it will be possible to generate multi-touch interactions at the point where they are placed. It can also be used as an automatic operation of game applications or as a command input aid. Since the proposed device can operate the touch panel in the same way as a person physically touches the touch panel, it is possible to operate and debug multiple devices in parallel at the same time, or to physically operate devices that are not connected to the network for security reasons by attaching the proposed wireless device. There is a product that tests the responsiveness of touch panels [7]. It would be able to generate multi-touch by this product with two nibs, but the proposed method is more useful because it can be downsized, does not need to be fixed, and can be embedded in the objects used in daily life.

The proposed device consists of an electrode sheet printed with electrodes using conductive ink and a voltage control board, and generates multi-touch interactions by changing the capacitance of the touch panel in time and space. Preliminary experiments were performed to investigate the size and spacing of the electrodes required to generate multi-touch interactions, and based on that the device was designed and implemented. In the evaluation experiments, it was confirmed that the proposed device can generate six multi-touch interactions of tap, press-and-tap, swipe, pinch-in, pinch-out, and rotation with high accuracy. We also measured the number of pixels that the generated swipe moved the page (movement), the rate at which the page was enlarged by pinch-out (enlargement), the rate at which the page was reduced by pinch-in (downsizing), and the angle at which the page was rotated by rotation (rotation). In addition, we investigated the minimum required interval until the next multi-touch interaction after the generation of the multi-touch interaction, and confirmed that the proposed device can generate multi-touch interactions at high speed.

This paper is organized as follows. Section 2 introduces related works. The proposed method is explained in Section 3 and device implementation is explained in Section 4. The evaluation experiment is detailed in Section 5. Application examples of the proposed method are described in Section 6. Section 7 concludes this paper.

2. Related Work

This section introduces the works on a tangible interface that enables interaction between human and computer through the touch panel and the works on touch interaction generation device.

2.1. Tangible Interface for Touch Panel

Capacitive touch panels recognize a touch by detecting the approach of a conductor such as the human body. This basic principle was proposed by Rekimoto [8] for SmartSkin, and since his proposal, many studies on the interface used on touch panels have been made. This section introduces works related to interfaces to touch panels.

Ikematsu et al. [9] proposed an interaction technique to extend the input area of capacitive touchpads and touchscreens through objects. A physical bar can be turned into a touch bar by touching the trackpad and the line width can be continuously changed according to the force of holding the pen in a gripping pen that incorporates a pressure-sensitive sensor. Bianchi et al. [10] proposed a widget that electrically connects the contact surface of a touch panel with a conductive material. It is a simple physical aid for active reading on a tablet computer that runs on a tablet device and runs without batteries. Clip-on Gadgets proposed by Yu et al. [11] use a clip with physical buttons attached to a touch panel such as a smartphone to perform touch input. Conductive rubber is attached inside the clip, and when a button is pressed, the capacitance of the conductive rubber changes to generate touch input. Touch input can be performed only at the place where the conductive rubber is arranged.

CapStones of Chan et al. [12] have multiple conductive parts with a block-like interface. By stacking CapStones on a touch panel, the number of conductive parts that touch the touch panel increases, creating different multi-touch patterns. In CapStones, the connection state of an object is obtained by a multi-touch pattern. GaussBricks proposed by Liang et al. [13] is a device embedded with a magnet and can detect the method of device connections by the Hall sensor in the touch panel. It is also possible while floating over the touch panel. In addition, an application that displays objects on the touch panel according to the shape of the connected interface is implemented. Jansen et al. [14] proposed a tangible interface for operating a large display by placing a slider or dial on a tablet.

Schmitz et al. [15] have proposed a design system for 3D objects with capacitive touch sensors, and based on this, a tangible interface with conductive parts using a 3D printer has been proposed [16,17]. Kato et al. [16] have proposed a tangible interface that generates touch input on a touch panel when the user touches a grid with multiple conductive points by printing it with a fused deposition modeling (FDM)-based 3D printer. This allows the user to control the touch input at any position on the printed interface surface. CapWidgets proposed by Kratz et al. [17] use a knob-type interface with two conductive parts arranged on the bottom of its surface, turning the knob changes the capacitance of the conductive part and can operate the touch panel.

As introduced above, there have been proposed many tangible interfaces that produce touch interaction on touch panels. By touching the tangible objects, predefined touch pattern(s) are invoked on a touch panel, while our method enables multiple dynamic touch inputs, such as swipe and pinch-in, by controlling the capacitance change points.

2.2. Touch Interaction Generation Device

Table 1 shows the methodology, performance, and feature of the devices in the related works on touch interaction generation and our device for the comparison at a glance. The marker fabricated by Yu et al. [18] changes the capacitance between the touch panel and the marker by applying an electric current to generate a touch on the touch panel. Ogata et al. [19] have realized data transmission on a touch screen of a multi-touch device using a pen-type device that can electronically change the judgment of touch sensing. Kim et al. [20] have built a mechanism called SikuliBot that uses a 3D printer to physically move a stylus pen and input data to a touch panel. Voelker et al. [21] created a multi-touch interface that does not require contact with a battery or human body by electrostatically coupling a transmitting electrode in a touch panel to a receiving electrode at another position via a conductive bridge as an interface for generating touch input. Takashina et al. [22] have proposed a method to operate the touch panel without using fingers to automate the operation of the touch panel. They propose two methods to achieve the quasi-touch: the resistance control method, in which a semiconductor switch is contacted to cause a direct charge movement on the surface, and the capacitance sensor control method, in which an alternating-current (AC) signal is given from the outside to deceive the capacitive sensor. In this study, the quasi-touch is realized by the same method as the resistance control method.

Kawahara et al. [23] have proposed a method to generate conductive patterns using a multifunction machine and conductive ink. The followings works use conductive inks. Ikeda et al. [24] proposed a CapacitiveMarker that can generate a multi-touch by touching a two-dimensional electrode pattern printed with conductive ink on the touch panel. This can be applied to viewpoint switching in a map application or character switching in a game. Wiethoff et al. [25] created Sketch-a-TUI, which creates a different multi-touch pattern for each object by drawing conductive parts on a paper object with conductive ink, and identifies the object using the multi-touch pattern. Yamanaka et al. [26] proposed a SheetKey that generates complex touch input patterns printed with conductive ink for secure authentication against shoulder surfing. ExtensionSticker [6] by Kato et al. is an interface to extend the touch panel by sticking it to the touch panel with a striped seal made using conductive ink. By touching the conductive part of the sticker, a touch input corresponding to the print pattern is generated on the touch panel. Ikematsu et al. [27] have proposed LightTouch for

dynamically generating multi-touch. LightTouch is a photoresistor-based passive interface for converting visible light into capacitive touch inputs. It simulates finger operations such as tapping, swiping, or multi-touch gestures using conductive materials and photoresistors embedded inside the objects.

Table 1. Method, performance, and features of the related works on touch interaction generation. Note that dynamic multi-touch means multi-touch interaction with motion such as swipe and static multi-touch means multi-touch pointing without motion.

Ref.	Methodology	Performance	Features
Ours	Controlling the grounding state of printed electrodes using relays.	Multi-touch interactions at 88–100% accuracy.	Multiple dynamic multi-touch available; touchpoints can move; battery required
TUIC [18]	Fabricated conductive marker changing the capacitance on the touch panel by applying an electric current.	Half-cycle input interval is 12 ms.	Single static multi-touch available; touchpoints cannot move; battery required
FlashTouch [19]	Controlling the grounding state of conductive sponge using a relay.	Transmit data via a single touch pattern at 33 bps.	Single-touch available; touchpoints cannot move; battery required
SikuliBot [20]	Controlling a stylus attached to a 3D printer extruder with computer vision.	Single-touch interactions can be generated like a human.	Any single-touch available; touchpoints can move; battery required; big device
PUCS [21]	Electrostatically coupling a transmitting electrode with a receiving electrode in a touch panel via a conductive bridge.	90% or more touch detection accuracy; marker size: 6–10 mm; bridge length: +20 mm.	Static multi-touch available; touchpoints cannot move; no battery required
Device of Takashina et al. [22]	A variable resistance matrix to the ground on a specified point on the touch panel.	Touch at up to 3 points are generated.	Static multi-touch available; touchpoints cannot move; battery required
Capacitive Marker [24]	A 2D electrode pattern printed with conductive ink.	A static multi-touch pattern recognized in 19 ms.	Single static multi-touch available; touchpoints cannot move; no battery required
Sketch-a-TUI [25]	Drawing parts on a paper object with conductive ink.	Objects are identified on a touch panel from conductive pattern. No evaluation.	Single static multi-touch available; touchpoints cannot move; no battery required
SheetKey [26]	Printing the touch input patterns on the sheet with conductive ink.	The user slides a finger on the sheet and authenticated with 78–87% accuracy.	Single dynamic multi-touch available; touchpoints cannot move; no battery required
Extension Sticker [6]	Printing stripe pattern on the sheet with conductive ink.	Touching the conductive part, a touch input corresponding to the print pattern is generated.	Single dynamic multi-touch available; touchpoints can move; no battery required
LightTouch [27]	Change grounding state of electrodes using photoresistor and visible light.	Touch, swipe, and pinch-in/out can be generated. No evaluation.	Touch, swipe, and pinch-in-out available; rotation not available

For the touch generation methods introduced in the related studies, it is possible to generate predefined multi-touch patterns or mimic and transform finger touch inputs. In this paper we propose a device that generates multi-touch interactions using a sheet on which electrodes are printed with conductive ink. Unlike conventional works, in this paper a single device can dynamically generate multiple multi-touch interactions. In addition, LightTouch [27] needs to install the application to the PC of the touch panel that emits visible light through the display, while our device does not need any software installation or setup. The idea of the proposed method and preliminary experiments were presented

at the conference adjunct proceedings paper [28]. In this study, a new prototype was implemented, and a more detailed evaluation was conducted to validate the effectiveness.

3. Proposed Method

This section explains the configuration and processing of the proposed method.

3.1. Touch Recognition Mechanism of General Capacitive Touch Panel

A touch detection method in a capacitive touch panel is briefly explained. Please refer to the literature [8,29,30] for details. The capacitive touch panel senses the change in capacitance by the capacitance coupling effect [29]. There are two types of capacitive touch panels, surface capacitive and projected capacitive, but currently, they are mainly made with projected capacitive technology (PCT) [30]. In a projected capacitive touch panel, a signal transmitting electrode and a signal receiving electrode are arranged in a grid pattern, and electrostatic coupling occurs at the intersection of the transmitting and receiving electrodes. When a grounded conductor, such as the human body, approaches a place where electrostatic coupling is performed, the capacitance between the electrodes changes. The touch is detected by measuring the amount of change in the capacitance. Multiple touches can be detected by scanning the signal of the transmission electrodes.

Since the touch panel adapts to the operations with fingers, touch is not detected if the area where the capacitance is changing is too small or too large. In addition, a threshold is set for the change in capacitance so that the touch is not detected before the finger touches the touch panel. Furthermore, when two points are touched, if the distance between the two points is shorter than the grid spacing of the electrodes, the two points cannot be separated and recognized as a tap. In this paper, we create a device that takes these constraints into account.

3.2. Multi-Touch Interactions

In the proposed device, the electrodes are grounded under the control of a microcontroller and a relay, and eight types of multi-touch interaction: tap, double-tap, long-press, press-and-tap, swipe, pinch-in, pinch-out, and rotation, are realized. Each multi-touch interaction is shown in Figure 1.

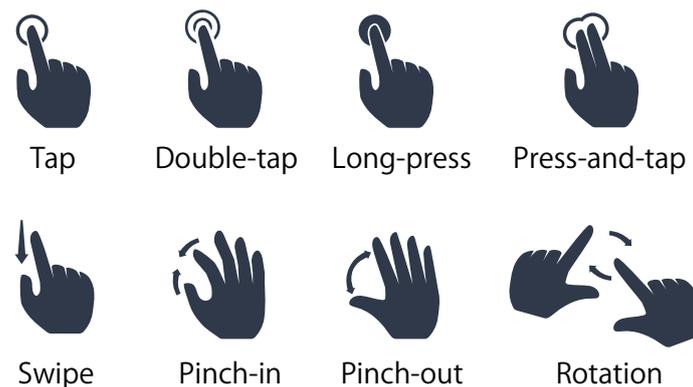


Figure 1. Illustrations of multi-touch interactions (designed by Macrovector—Freepik.com (accessed on 18 June 2021)).

Tap is a single-touch operation and is equivalent to a left-clicking with the mouse. Double-tap is a quick two-touch operation, and is equivalent to a double-click. Long-press is an operation that keeps touching and is equivalent to right-clicking with the mouse. Long-press is also known as press-and-hold. Press-and-tap is a touch operation with one finger while touching with the other finger, which is also equivalent to right-clicking. Both long-press and press-and-tap are equivalent to right-clicking a mouse, however long-press requires touching the touch panel for a few seconds, while press-and-tap allows quick right-clicking. Swipe is the operation of moving in one direction while touching with a

finger, and is equivalent to moving the mouse wheel. Pinch-in is an operation of pinching with two fingers, and is equivalent to downsizing the object on the screen by moving the mouse wheel while pressing the Ctrl key. Pinch-out is an operation to move two fingers in opposite directions while touching with the fingers and is equivalent to enlarging the object on the screen by moving the mouse wheel while pressing the Ctrl key as well as pinch-in. Rotation is the operation of touching with one finger as a center and moving the other finger on the touch panel similar to drawing a circle around the center finger, which is equivalent to rotating an object in the screen. Rotation cannot be realized with a mouse.

3.3. Multi-Touch Interaction Generation Method

The proposed method uses an electrode sheet made by printing electrodes on a plastic film using conductive ink. The method of realizing each touch interaction is as follows. Tap ground one electrode only once. Double-tap generates two taps quickly. Long-press grounds one electrode for a certain time. During the electrode is grounded, long-press is effective.

The remaining five types of interactions are generated by continuously grounding multiple electrodes and moving the center point of the change in capacitance. The procedure for generating a swipe is shown in Figure 2. In the figure, ■ (black square) is the grounded electrode, □ (white square) is the ungrounded electrode, and the blue circle is the center point of the capacitance change recognized by the touch panel. The procedure for generating the swipe is as follows.

1. Electrode A is grounded and the center of the capacitance change is generated at the position of electrode A.
2. Electrode B is also grounded and the center of the capacitance change is moved to the midpoint of the two electrodes.
3. Electrode A is released (not grounded), and the center of the capacitance change is moved to the position of electrode B.

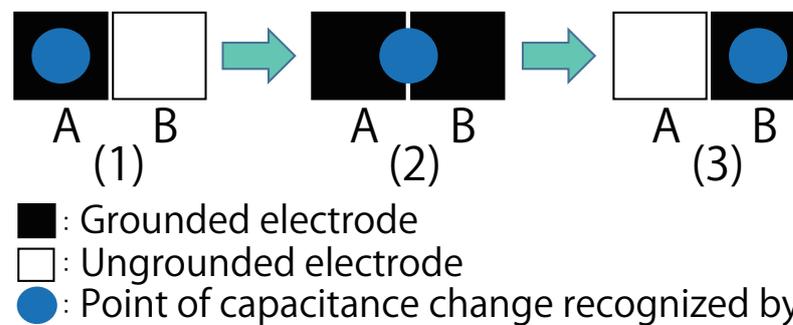


Figure 2. Generation procedure of swipe.

This procedure is carried out with two electrodes adjacent to each other in the swipe direction to generate a swipe. This method takes advantage of the constraint that if the distance between adjacent electrodes is shorter than the grid electrode spacing inside the touch panel, two points of grounding are misidentified as one point of touch.

Other interactions are realized by a combination of tap, long-press, and swipe as shown in Figure 3. In the figure, ■ (black square) is the grounded electrode, □ (white square) is the ungrounded electrode, and the blue circle is the center point of the capacitance change recognized by the touch panel. Press-and-tap keeps one electrode grounded while another non-adjacent electrode is grounded. Pinch-in/-out is created with two swipes in the direction of approaching/away. Rotation long-press one electrode and creates swipe at non-adjacent positions.

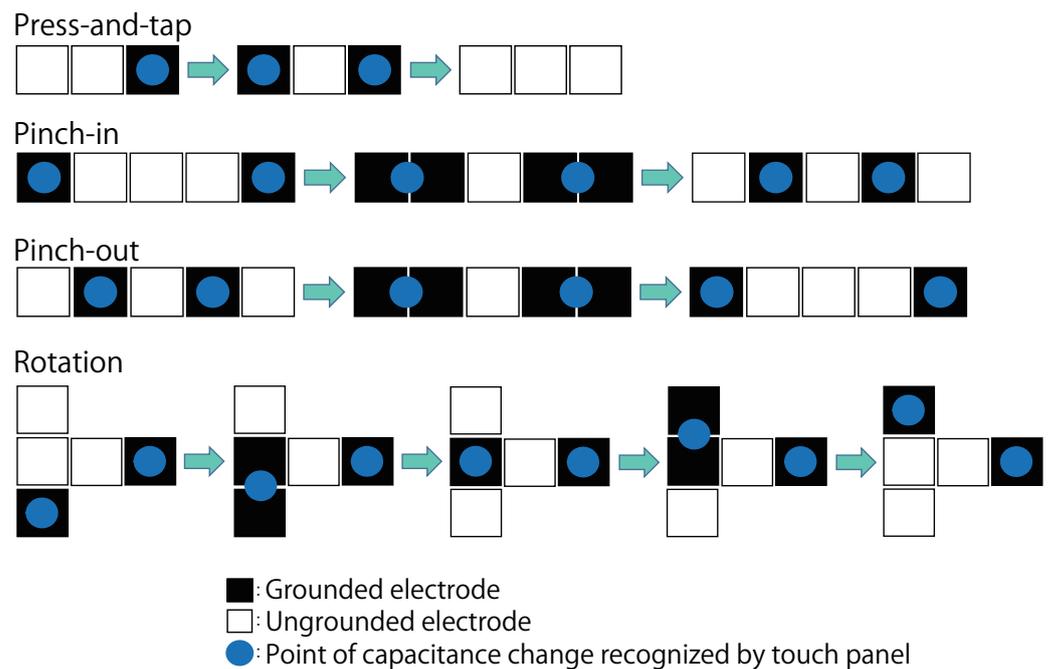


Figure 3. Generation procedure of press-and-tap, pinch-in, pinch-out, and rotation.

In addition to these, there are drag and flick interactions. Drag is generated with a long-press and then swipe, and we confirmed that drag can be generated in this way. We also confirmed that flick can be generated by generating a swipe whose last electrode is grounded for short time.

In the touch generation methods of related studies, it is possible to generate unique multi-touch patterns or to output a finger touch input as a separate touch input using a sheet with printed conductive parts. However, it is necessary to combine multiple interfaces or design new interfaces to generate different touch patterns. The proposed method can dynamically generate various multi-touches with a voltage control board, a battery, and a single electrode sheet.

4. Implementation

This section describes a preliminary experiment to investigate the electrode size required for multi-touch interaction generation and an implementation based on the results of the preliminary experiment.

4.1. Preliminary Experiment

According to the literature [21], it is assumed that the grid electrode spacing inside the touch panel is generally 5 mm. The longest distance of the diagonal electrode intersections is $5\sqrt{2} \approx 7.07$ mm. If the distance between electrodes of our device is shorter than this, multi-touch interaction may not be generated. On the other hand, if the electrodes are distant, the device becomes large and the interaction generation in the procedure of Figures 2 and 3 results in separate taps, so the electrode spacing should be the smallest one that generates multi-touch interaction with high accuracy. Therefore, a prototype device was built and its success rate, i.e., the number of success input per trials, was investigated for four interactions that move the center of the capacitance change point: swipe, pinch-in, pinch-out, and rotation, at different electrode sizes.

The electrode sheet was made by printing electrodes on a plastic film (NB-TP-3GU100 manufactured by Mitsubishi Paper Mills, Ltd., Tokyo, Japan) using conductive ink (silver nano-particle ink manufactured by Mitsubishi Paper Mills) by an inkjet printer (PX-S160T manufactured by EPSON) as shown in Figure 4. In this study, conductive ink was used, but technology to print electrodes using a 3D printer has been proposed [31], which is also the other solution to realize our device. One electrode was made of squares of five

sizes with sides of 4, 5, 6, 7, and 8 mm, and the gap between the electrodes was set to 0.5 mm. The 0.5 mm gap is the minimum distance that the two electrodes do not carry voltage. When the electrode size is determined, the gap is set to 0.5 mm, and the electrode distance is automatically determined, so the evaluation is conducted by changing only the electrode size. The touch panel used was a Lenovo ThinkPad X1 Yoga (14 inch, 30.94 cm width, 17.42 cm height display; Windows 7; 10-point multi-touch compatible). The display resolution is set to 1920×1080 pixels and divided by side length; density is approximately 157 pixels per inch (ppi). Each multi-touch interaction was generated 100 times using five different sized electrode sheets in two directions: parallel and diagonal (tilted 45 degrees to the parallel direction) to the grid electrode inside the touch panel, for a total of 4 types of multi-touch interaction \times 5 types of sheets \times 2 directions \times 100 = 4000 times and the success rate was measured, e.g., the success rate is 90% when 90 out of 100 trials are successfully input. Generation in the parallel direction was triggered by placing the rectangular electrode sheet parallel to the edge of the touch panel, while generation in the diagonal direction was triggered by placing the rectangular electrode sheet at a 45-degree angle to the edge of the touch panel.

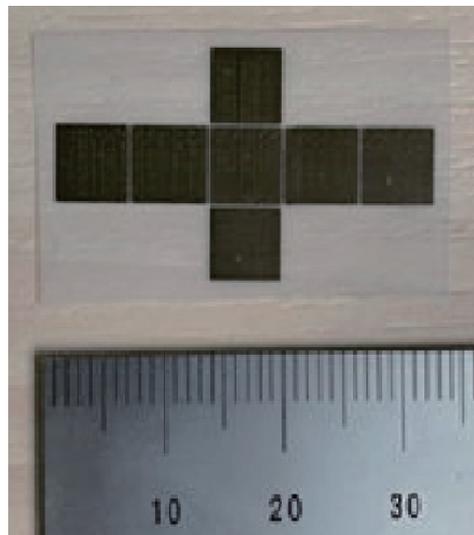


Figure 4. Electrode sheet used in the preliminary experiment. The scale is millimeters.

We implemented an application in C# that obtains touch input information. As a means of detecting touch input, we use `ManipulationDeltaEvent`, which can acquire the amount of movement, downsizing, enlargement, and rotation of an object when a touch operation is performed on an object placed in a windows application. Therefore, we placed a square object in the application and generated touch input on that object. More specifically, `ManipulationStartingEvent` is triggered when a touch input starts (i.e., when a finger touched the panel), `ManipulationDeltaEvent` is triggered when the touch input moves, and `ManipulationInertiaStartingEvent` is triggered when the touch input ends (i.e., when a finger leaves the panel). Through the events, position, amount of downsizing/enlargement, and amount of rotation can be obtained. The success of each multi-touch interaction is judged as follows. For swipe, if the positions of the object at the end of touch input and the start of touch input are not identical, the swipe is counted as a success. For pinch-in, if the amount of scale ratio at the end of touch input in proportion to the original object size is less than 1, the pinch-in is counted as a success. For pinch-out, if the amount of scale ratio at the end of touch input in proportion to the original object size is more than 1, the pinch-out is counted as a success. For rotation, if the amount of rotation at the end of touch input compared to the original object orientation is not zero, the rotation is counted as a success.

The duration of grounding and ungrounding the electrodes is explained with an example of the time series of controlling two electrodes A and B to generate a swipe as

shown in Figure 5. As seen in the figure, the time to switch between the grounding and ungrounding of the electrodes is called switching time in this paper, and all switching time is set to 50 ms in the preliminary experiments. More specifically, one snapshot of switching pattern in an interaction remains for a switching time. The time between the end of one multi-touch interaction and the start of the next multi-touch interaction is called the interval and was set to 1000 ms. These values are enough long to generate and secure the interaction generation so that the electrode sizes can be investigated. The investigation of switching time and the success rate is conducted in the later evaluation section.

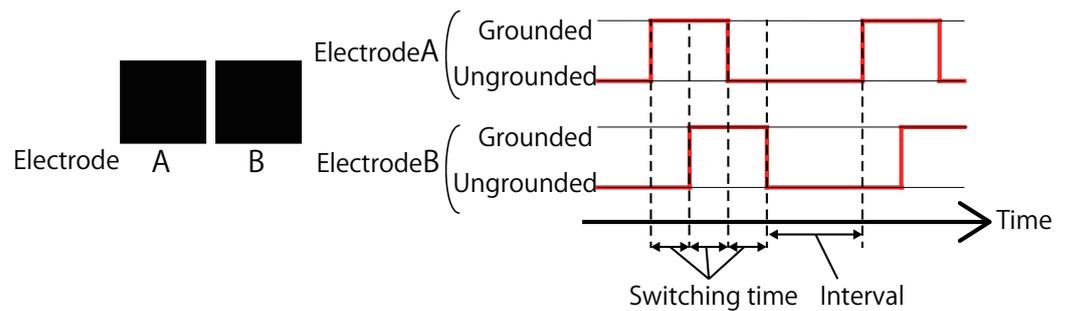


Figure 5. An example of time series for controlling electrodes to generate swipe.

The results are shown in Table 2. The success rate of the swipe was 100% for electrode size more than 7 mm in both the parallel and diagonal directions. These results are generally in line with our expectations that the grid electrode spacing inside the touch panel is generally 5 mm and distance of the diagonal electrode intersections is $5\sqrt{2} \approx 7.07$ mm.

Table 2. Success rate of interaction generation [%]. (Diag. = Diagonal).

Electrode Size	Swipe		Pinch-In		Pinch-Out		Rotation	
	Parallel	Diag.	Parallel	Diag.	Parallel	Diag.	Parallel	Diag.
4 mm	9	24	27	75	61	65	97	94
5 mm	95	93	71	84	55	98	100	100
6 mm	98	97	100	100	100	100	100	99
7 mm	100	100	100	100	100	100	100	100
8 mm	100	100	100	100	100	100	100	100

Other multi-touch interactions could be generated with a 5 mm electrode. While the swipe is generated by two adjacent electrodes, the pinch-in and pinch-out electrodes use five electrodes and the swipe part in the rotation uses three electrodes, and the success rate is estimated to be higher even for small electrodes due to the large displacement of the center of the capacitance change. In the 4 mm experimental conditions, the success rate of swipe was lower, even though complex operations such as Pinch-in/out can be recognized. The complex operations invoked electrodes in a wider area than swipe, which might lead to higher success rate. The electrode size should be at least 7 mm to generate swipe with high accuracy.

4.2. Device Implementation

Based on the results of preliminary experiments, we designed a honeycomb structure with nine regular hexagonal electrodes with a distance of 7 mm to opposing surfaces across a flat surface to minimize the printed area of the electrodes and to increase the granularity in the direction of the swipe. In Figure 6, \bullet (black hexagon) is the grounded electrode, \square (white hexagon) is the ungrounded electrode, and the blue circle is the center point of the capacitance change recognized by the touch panel. A swipe using two adjacent electrodes can be generate in six directions $\pi/3 \times n$ [rad] ($n = 0, 1, 2, 3, 4, 5$), and a swipe using three electrodes can be generated in different six directions $\pi/6 + \pi/3 \times n$ [rad] ($n = 0, 1, 2, 3, 4, 5$), enabling 12 directions in total. In Pinch-in and pinch-out were

achieved by generating two sets of swipes with four electrodes and rotation was achieved by switching the electrodes to draw two arcs simultaneously with six electrodes. Tap, double-tap, long-press, and press-and-tap are the same as in the preliminary experiment as these do not require movements.

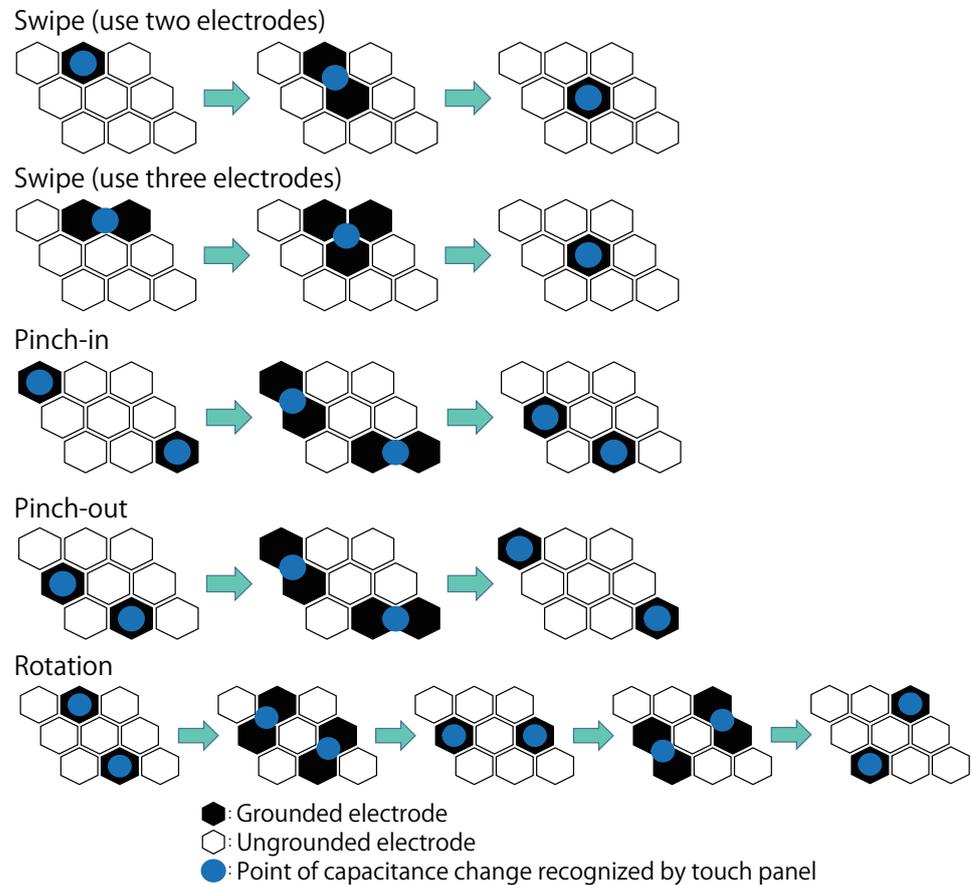


Figure 6. Multi-touch interaction generation procedure in the honeycomb structure of the proposed device.

Figure 7 shows the configuration of the proposed device. The device consists of a one-board microcontroller, a circuit consisting of a mechanical relay, an electrode sheet, and a power supply. Multi-touch interactions are generated by pre-programming the electrode control pattern. The circuit diagram is shown in Figure 8. An Arduino Uno was used as the microcontroller. The electrodes and microcontroller were connected via a relay (946H-1C-5D). The relay and the electrode are connected by a conducting wire, and the electrodes and the conducting wires are bonded by conductive tape. The microcontroller is powered by universal serial bus (USB).

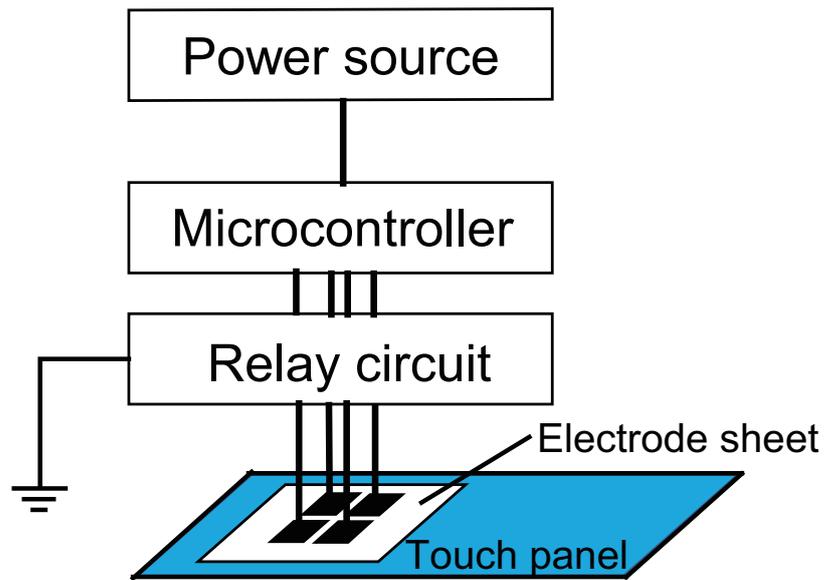


Figure 7. Device configuration.

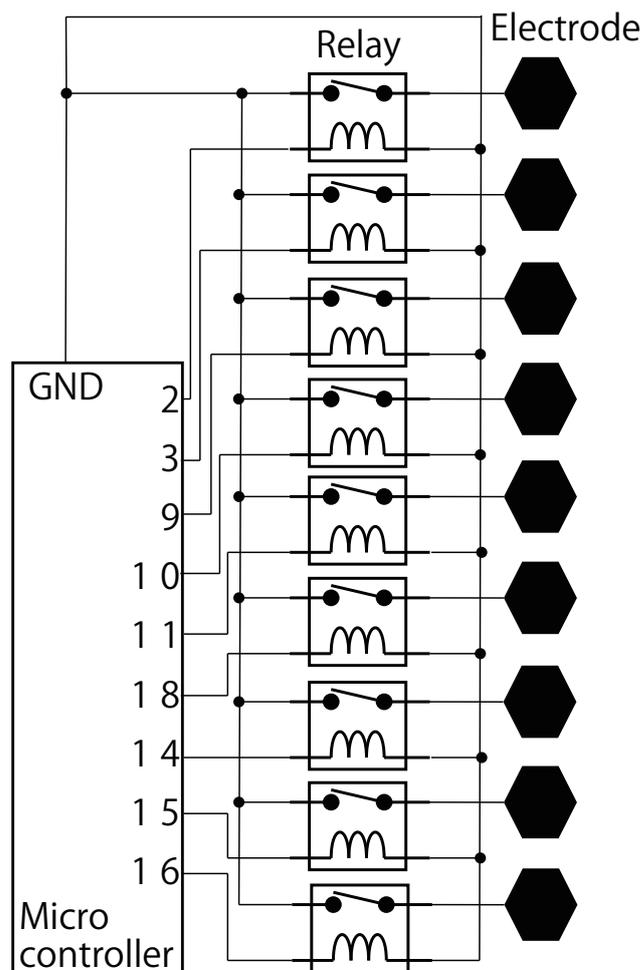


Figure 8. Device schematic.

5. Evaluation

In this section, we evaluate the success rate of multi-touch interactions and the amount of swipe, pinch-in, pinch-out, and rotation.

5.1. Success Rate of Multi-Touch Interactions

5.1.1. Evaluation Environment

If a single multi-touch creation takes several seconds, it cannot be used as an interface, and it is preferable to be able to create interactions on the display within a short time after sending the creation command. In some cases, swiping and zooming multiple times in succession can result in long moves and large zooms, so it is desirable to be able to generate a large number of touch interactions in a short period of time. Therefore, we evaluate the success rate for six interactions with different switching times: tap, press-and-tap, swipe, pinch-in, pinch-out, and rotation by changing the switching time. Double-tap is omitted from the evaluation since we have confirmed that it is generated by two taps with short intervals. Long-press is also omitted from the evaluation since it is just grounding an electrode for a longer time than tap. If only taps were performed, it would simply be an evaluation of the responsiveness of the touch device, but we evaluated our device because it is unknown whether the success rate of interactions with motion, such as swipe, pinch-in, pinch-out, and rotation, is similar to that of taps.

The touch panel used was a Lenovo ThinkPad X1 Carbon (14 inch, 30.94 cm width, 17.42 cm height display; Windows 10; 10-point multi-touch compatible). The display resolution is set to 1920×1080 pixels and divided by side length; density is approximately 157 ppi. We obtain touch input information by the same application as in the preliminary experiment in Section 4.1. Figure 9 shows the experimental setup of Arduino Uno, mechanical relays, electrode array, and a touch panel. The electrodes are connected to the relays via enamel coated cable and the relays are connected to the pins of Arduino Uno. The operation is sent from the laptop. The C# application run on the laptop acquires the amount of movement, downsizing, enlargement, and rotation of an object when a touch operation is performed on a red square object placed in a windows application. The electrodes generate touch input on that object. The success of tap and press-and-tap is determined as follows. For tap, when ManipulationInertiaStartingEvent is triggered if the amount of scale ratio and rotation are unchanged, the tap is counted as a success. For press-and-tap, when the second tap is detected while the first press is being detected, the press-and-tap is counted as a success.

For tap, the switching time was changed from 1 ms to 25 ms in steps of 1 ms and from 25 ms to 200 ms in steps of 5 ms. For each switching time, taps were generated 100 times continuously without an interval (6000 times in total). More specifically, when switching time is 5 ms, the electrode is grounded for 5 ms, ungrounded for 5 ms, grounded for 5 ms, and to be the same afterward. Except for the tap, the switching time was changed from 5 ms to 200 ms in steps of 5 ms. For each switching time, the interval was fixed to 500 ms. For each switching time, each interaction was generated 100 times, 4000 times in total.

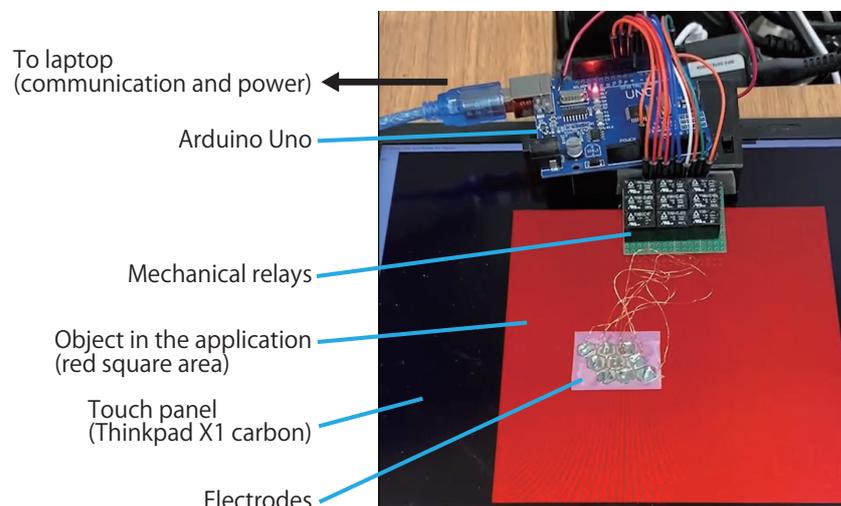


Figure 9. Experimental setup of the microcontroller, relays, electrodes, and laptop in the evaluation.

5.1.2. Results

The success rates for tap, press-and-tap, swipe, pinch-in, pinch-out, and rotation are shown in Figure 10. In the following, the results are explained for each interaction.

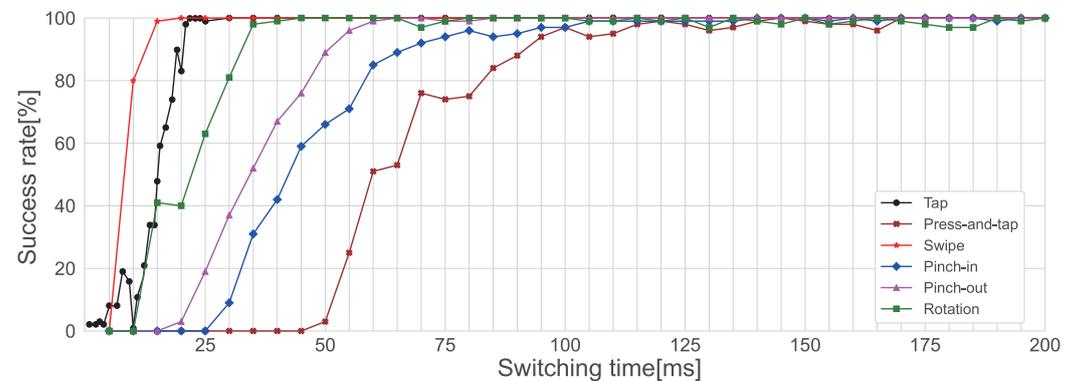


Figure 10. The success rate of multi-touch interaction generation vs. switching time.

Tap was generated with a success rate of almost 100% when switching time was equal to or more than 22 ms. When the switching time was less than 15 ms, the success rate was less than 50%. When the tap generation failed, any other false interactions were not generated. Press-and-tap was generated with a success rate of 100% when the switching time was equal to or more than 170 ms and 94% or more when switching time was equal to or more than 95 ms. The success rate was 0% when the switching time was equal to or less than 45 ms. When the press-and-tap generation failed, mostly single tap was incorrectly generated and the rest are no interactions generated. Swipe was generated with a success rate of 100% when switching time was equal to or more than 20 ms, 99% when switching time was 15 ms, 80% when the switching time was 10 ms, and 0% when the switching time was 5 ms. When the swipe generation failed, mostly tap was incorrectly generated and the rest are no interactions generated. Pinch-in was generated with a success rate of 99% or more when the switching time was equal to or more than 105 ms and 0% when the switching time was less than 25 ms. When the pinch-in generation failed, mostly no interactions were generated and the rest are tap, swipe, or pinch-out incorrectly generated. Pinch-out was generated with a success rate of 99% or more when the switching time was equal to or more than 60 ms and 0% when the switching time was less than 15 ms. When the pinch-out generation failed, mostly no interactions were generated and the rest are tap or swipe incorrectly generated. Rotation was generated with a success rate of 99% or more when the switching time was more than 40 ms and 0% when the switching time was less than 10 ms. When the rotation generation failed, mostly no interactions were generated and the rest are tap or swipe incorrectly generated.

5.1.3. Discussion

The results show that tap can be generated at 25 touches per second with a success rate of almost 100% since 20 ms switching time means that one tap takes 40 ms. The success rate of tap significantly drops at a switching time of 20 ms or less, showing that the touch recognition of the touch panel is performed at intervals of approximately 20 ms, or there is a restriction in the software that the touch whose occurrence interval is less than 20 ms is not recognized. Particularly the result for tap in the short switching time (less than 10 ms) is not stable and may not be reliable due to the operation time of the relay used, which is less than 6 ms according to the datasheet. On the other hand, the success rate of longer switching time (more than 10 ms) is improving as switching time becomes longer. As the touch panel does not react for the switching time less than 20 ms, the operation time of the mechanical relay does not affect the performance of the proposed method.

5.2. Amount of Movement, Downsizing, Enlargement, and Rotation

5.2.1. Evaluation Environment

For multi-touch interactions with motion, such as swipe, pinch-in, pinch-out, and rotation, it is an important indicator that the amount of movement by swipe, amount of downsizing by pinch-in, amount of enlargement by pinch-out, and amount of rotation by rotation are the intended amount and stable amount. We measured the swipe movement, pinch-in downsizing, pinch-out enlargement, and rotation degrees for the successfully generated touch interactions in Section 5.1.

The amount of movement was recorded by calculating the distance between the positions of the object at the start of touch input and the end of touch input. Since the object moves in both the x-axis and y-axis directions, we calculated the square root of the sum of squares of the amount of movement in the x-axis direction and the amount of movement in the y-axis direction. The amount of downsizing and enlargement is the amount of scale ratio at the end of touch input to the start of touch input. `DeltaManipulation.Scale` is a property that can be retrieved when `ManipulationDeltaEvent` occurs, and the scale ratio of an object by touch input can be acquired. Similarly, the amount of rotation of an object by touch input can be acquired through `DeltaManipulation.Rotation`. Since `ManipulationDeltaEvent` occurs multiple times during a touch input, the amount of scale ratio was calculated as the product of `DeltaManipulation.Scale` from the start to the end of the touch input. The amount of rotation was calculated as the sum of `DeltaManipulation.Rotation` from the start to the end of the touch input.

5.2.2. Results and Discussion

The amounts of movement, downsizing, enlargement, and rotation are shown in Figures 11–14, respectively. In the figures, the blue line shows the success rate, red circles with black error bars show the average value and the standard deviation. The average and standard deviation are calculated over the successful input, therefore error bars are not shown for the cases of 0% success rate. In the amount of downsizing and enlargement, it represents the ratio of the object size after the interaction compared with the original size, so 100% means that the object size has not been changed.

For swipe, the average amount of movement was approximately 40 pixels (px) at almost of all switching times and the standard deviation increased as the switching time decreased. We have observed that the amount of movement often exceeded 100 px when the switching time was less than 20 ms. This may be due to the short grounding time of the electrode, which was recognized as a flick by the computer. The average amount of downsizing, enlargement, and rotation became smaller as the switching time became shorter. Please note that for the amount of downsizing and enlargement 100% means the original object size and that amount of downsizing and enlargement close to 100% means small change. For pinch-in, the average amount of downsizing was approximately 54% at 200 ms switching time, and the average amount of downsizing was 96% at 20 ms switching time. The standard deviation range was similar at all switching times. For pinch-out, the average amount of enlargement was approximately 184% at 200 ms switching time and 104% at 20 ms switching time. The standard deviation range was similar at all switching times. For rotation, the average amount of rotation was approximately 53 degrees at 200 ms switching time and 8 degrees at 20 ms switching time. The standard deviation range was similar at all switching times.

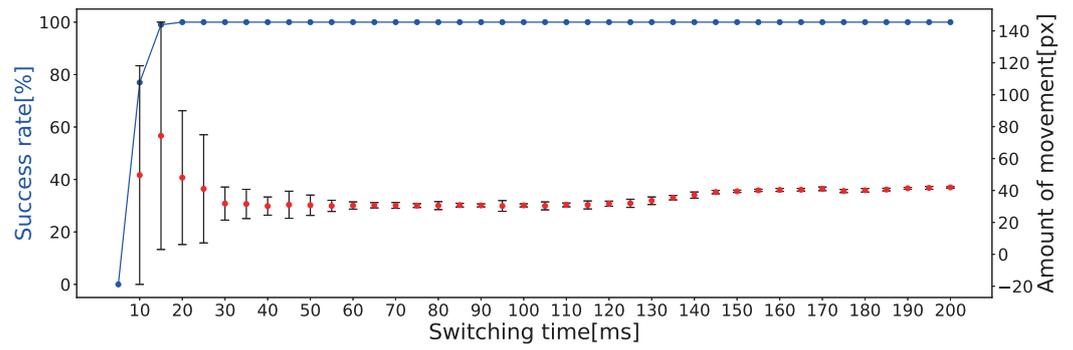


Figure 11. Amount of movement vs. switching time for swipe. Interval of interactions is fixed at 500 ms.

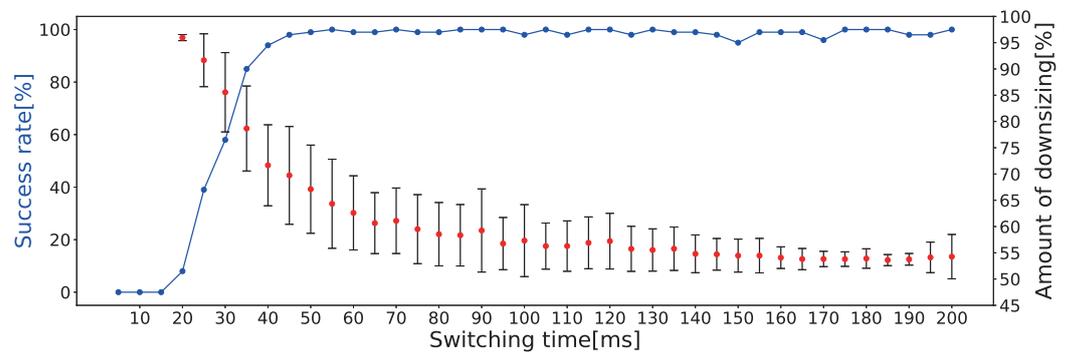


Figure 12. Amount of downsizing vs. switching time for pinch-in. Interval of interactions is fixed at 500 ms.

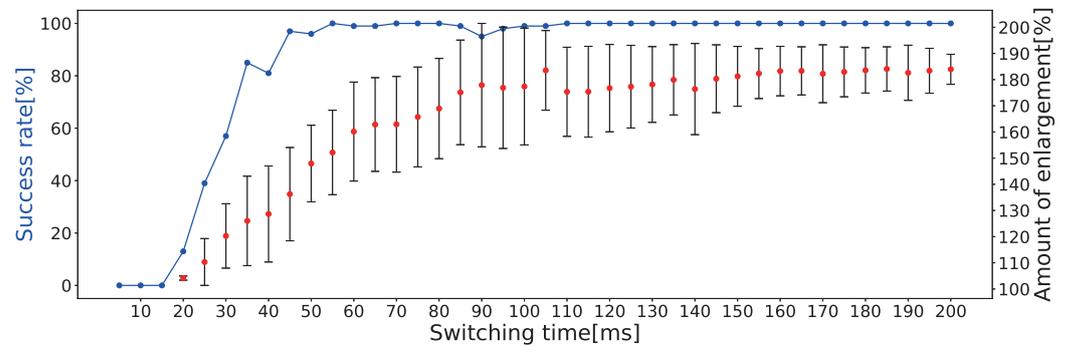


Figure 13. Amount of enlargement vs. switching time for pinch-out. Interval of interactions is fixed at 500 ms.

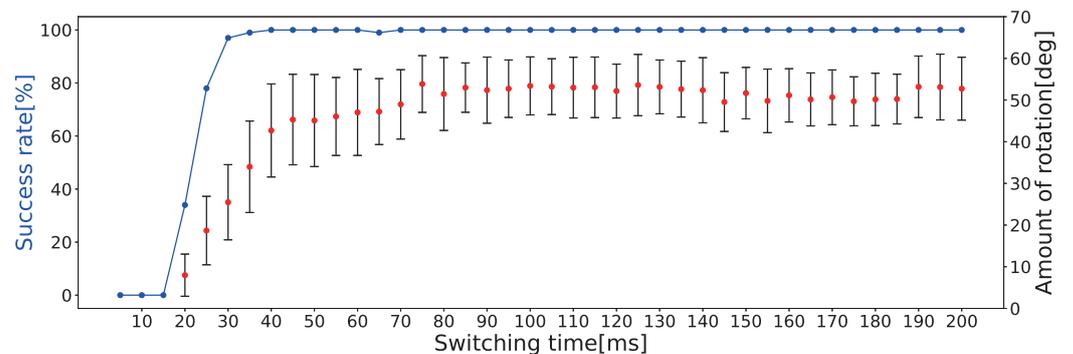


Figure 14. Amount of rotation degrees vs. switching time for rotation. Interval of interactions is fixed at 500 ms.

From these results, the average amount of swipe is stable with a long switching time and it would be possible to move an object in the required amount. On the other hand, it would be difficult to zoom-in, zoom-out, and rotate the object in the required exact amount with one interaction even by setting enough amount of switching time. The standard deviation of the enlargement at 20 ms switching time was small, which may have been due to the low success rate, i.e., there are a few successful interactions and there are no variety of result values. However, even humans cannot change the specified amount by the touch interactions. It is common to stop the interaction at an appropriate timing of the user seeing the object changed while performing the interaction, and we do not think that it is necessary to change the specified amount by one interaction. If the user wants to enlarge exactly 200% or rotate 90 degrees, it can be specified from the graphical user interface (GUI).

5.3. Interval of Interactions

The length of the multi-touch interaction interval affects the total operation time when generating multi-touch interactions in succession. To investigate the effect of the interval, the switching time was fixed at 100 ms and the interval varied from 5 ms to 100 ms in steps of 5 ms and the success rate was measured for four kinds of interactions: swipe, pinch-in, pinch-out, and rotation. Each interaction generated 100 times and 2000 times in total.

The result is shown in Figure 15. The success rate of a swipe was almost 100% when the interval was 55 ms or more, but it was less than 1% when 40 ms or less. The success rate of pinch-in was over 99% when the interval was 55 ms or more, but it was less than 6% when 40 ms or less. The success rate of pinch-out was 99% when the interval was 70 ms or more, but it was less than 2% when 45 ms or less. The success rate of rotation was 99% when the interval was 60 ms or more, but it was less than 1% when 40 ms or less. The success rate of all multi-touch interactions were significantly reduced below 60-ms interval, however, the success rate of all multi-touch interactions was 99% when the interval was 70 ms or more. Setting switching time to 100 ms and interval to 70 ms, a swipe takes 370 ms, pinch-in takes 370 ms, pinch-out takes 370 ms, and rotation takes 570 ms, which can be said that these multi-touch interactions can be produced within a reasonable time.

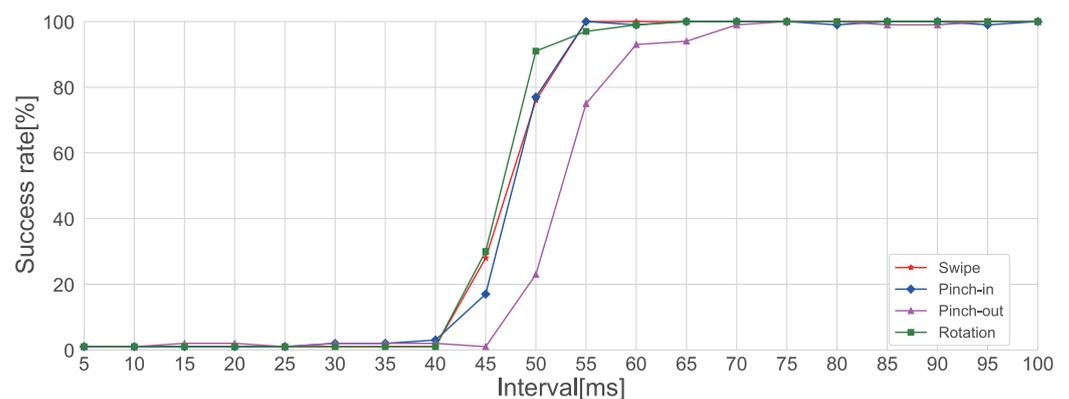


Figure 15. The success rate of multi-touch interaction generation vs. interval. The switching time is fixed at 100 ms.

6. Application

An example of the application using the proposed device is its implementation in a stylus pen. When using a stylus on a touch panel, touch interactions with fingers are difficult and messy to do while holding a stylus. Therefore, by implementing the proposed device in the pen tip, multi-touch interactions are realized without changing how to hold a stylus. For example, an object can be moved, rotated, zoomed-in, and zoomed-out while being drawn without detaching a stylus from the touch panel. In the current implementation, the size of the electrode sheet is about 3 cm squared, which is large

enough for a stylus for a tablet or laptop. On the other hand, the size of the stylus would be acceptable for a large touch panel for the educational, conference, and presentation uses, and we think it is useful to be able to draw, move, and rotate with a pen in one hand while explaining.

For another example, a touch input with fingers can be copied by the software of the touch panel, and the touch input can be reproduced with the proposed device. The proposed device can be used for a capacitive touch panel and can copy the touch input to another terminal without any software installation. Moreover, if there are multiple proposed devices, the similar touch input can be generated simultaneously on multiple touch panels, so that one touch panel can operate multiple touch panels, which would be useful for testing and debugging software with multiple different devices. The amount of movement may not be accurate but the success rate is almost 100% with the proposed method, enabling the check of the operation of the software. To cover the entire screen with the electrode sheet of the proposed device, many relays are required and a dedicated circuit board must be implemented, and parts other than the electrode sheet would be large. However, it is considered to be acceptable because it is not intended to be held by the hand at all times. Another advantage is that if the same operation is generated on multiple devices, only one relay circuit hardware is needed.

7. Conclusions

In this paper, we constructed a device that generates various multi-touch interactions using a printable circuit and a one-board microcontroller. Based on the results of preliminary experiments, a device with an appropriately sized electrode was created and the success rate of six types of multi-touch interaction generation—tap, press-and-tap, swipe, pinch-in, pinch-out, and rotation—was evaluated. Tap was generated at a rate of 25 times/s at 100%. Press-and-tap, swipe, pinch-in, pinch-out, and rotation were generated at a maximum of 100%. We also confirmed that high-speed continuous swipe can be generated with 99+% accuracy: swipe takes 370 ms, pinch-in takes 370 ms, pinch-out takes 370 ms, and rotation takes 570 ms. In addition, we also measured the amount of swipe movement, pinch-in downsizing, pinch-out enlargement, and rotation degrees, and although there was variation in their amounts, swipe movement was stable at approximately 40 px.

Author Contributions: Conceptualization, M.O. and K.M.; methodology, M.O. and K.M.; software, M.O.; validation, M.O. and K.M.; formal analysis, M.O. and K.M.; investigation, M.O.; resources, M.O.; data curation, M.O.; writing—original draft preparation, M.O.; writing—review and editing, K.M.; visualization, M.O.; supervision, K.M.; project administration, K.M.; funding acquisition, K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Japan Science and Technology Agency, PRESTO grant number JPMJPR1937, Japan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Klamka, K.; Horak, T.; Dachselt, R. Watch+Strap: Extending Smartwatches with Interactive StrapDisplays. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI 2020), Honolulu, HI, USA, 25–30 April 2020; paper No. 72.
2. Wang, B.; Grossman, T. BlyncSync: Enabling Multimodal Smartwatch Gestures with Synchronous Touch and Blink. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI 2020), Honolulu, HI, USA, 25–30 April 2020; paper No. 5.
3. Roudaut, A.; Lecolinet, E.; Guiard, Y. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In Proceedings of the 27th Annual ACM Conference on Human Factors in Computing Systems (CHI 2009), Boston, MA, USA, 4–9 April 2009; pp. 927–936.
4. Roth, V.; Turner, T. Bezel Swipe: Conflict-Free Scrolling and Multiple Selection on Mobile Touch Screen Devices. In Proceedings of the 27th Annual ACM Conference on Human Factors in Computing Systems (CHI 2009), Boston, MA, USA, 4–9 April 2009; pp. 1523–1526.
5. Yasu, K. Magnetact: Magnetic-sheet-based Haptic Interfaces for Touch Devices. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI 2019), Glasgow, UK, 4–9 May 2019; paper No. 240.

6. Kato, K.; Miyashita, H. ExtensionSticker: A Proposal for A Striped Pattern Sticker to Extend Touch Interfaces and its Assessment. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI 2015), Seoul, Korea, 18–23 April 2015; pp. 1851–1854.
7. Touch Performance Testing Solutions. Available online: <https://www.optofidelity.com/offering/products/touch> (accessed on 28 May 2020).
8. Rekimoto, J. SmartSkin: An Infrastructure for Freehand Manipulation on Interactive Surfaces. In Proceedings of the 20th Annual ACM Conference on Human Factors in Computing Systems (CHI 2002), Minneapolis, MN, USA, 20–25 April 2002; pp. 113–120.
9. Ikematsu, K.; Siiro, I. Ohmic-touch: Extending Touch Interaction by Indirect Touch through Resistive Objects. In Proceedings of the 36th Annual ACM Conference on Human Factors in Computing Systems (CHI 2018), Montréal, QC, Canada, 23–26 April 2018; pp. 521:1–521:8.
10. Bianchi, A.; Ban, S.R.; Oakley, I. Designing a Physical Aid to Support Active Reading on Tablets. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI 2015), Seoul, Korea, 18–23 April 2015; pp. 699–708.
11. Yu, N.; Tsai, S.; Hsiao, I.; Tsai, D.; Lee, M.; Chen, M.Y.; Hung, Y. Clip-on Gadgets: Expanding Multi-touch Interaction Area with Unpowered Tactile Controls. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST 2011), Santa Barbara, CA, USA, 7–12 October 2011; pp. 367–372.
12. Chan, L.; Müller, S.; Roudaut, A.; Baudisch, P. ZebraWidgets: Sensing Stacks of Building Blocks, Dials and Sliders on Capacitive Touch Screens. In Proceedings of the 30th Annual ACM Conference on Human Factors in Computing Systems (CHI 2012), Austin, TX, USA, 5–10 May 2012; pp. 2189–2192.
13. Liang, R.; Chan, L.; Tseng, H.; Kuo, H.; Huang, D.; Yang, D.; Chen, B. GaussBricks: Magnetic Building Blocks for Constructive Tangible Interactions on Portable Displays. In Proceedings of the Extended Abstracts of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI 2014), Toronto, BC, Canada, 26 April–1 May 2014; pp. 3153–3161.
14. Jansen, Y.; Dragicevic, P.; Fekete, J. Tangible Remote Controllers for Wall-Size Displays. In Proceedings of the 30th Annual ACM Conference on Human Factors in Computing Systems (CHI 2012), Austin, TX, USA, 5–10 May 2012; pp. 2865–2874.
15. Schmitz, M.; Khalilbeigi, M.; Balwierz, M.; Lissermann, R.; Mühlhäuser, M.; Steimle, J. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST 2015), Charlotte, NC, USA, 8–11 November 2015; pp. 253–258.
16. Kato, K.; Miyashita, H. 3D Printed Physical Interfaces that can Extend Touch Devices. In Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology (UIST 2016), Tokyo, Japan, 16–19 October 2016; pp. 47–49.
17. Kratz, S.; Westermann, T.; Rohs, M.; Essl, G. CapWidgets: Tangible Widgets versus Multi-Touch Controls on Mobile Devices. In Proceedings of the Extended Abstracts of the 29th Annual ACM Conference on Human Factors in Computing Systems (CHI 2011), Vancouver, BC, Canada, 7–12 May 2011; pp. 1351–1356.
18. Yu, N.H.; Chan, L.W.; Lau, S.Y.; Tsai, S.S.; Hsiao, I.C.; Tsai, D.J.; Hsiao, F.I.; Cheng, L.P.; Chen, M.; Huang, P.; et al. TUIC: Enabling Tangible Interaction on Capacitive Multi-touch Displays. In Proceedings of the 29th Annual ACM Conference on Human Factors in Computing Systems (CHI 2011), Vancouver, BC, Canada, 7–12 May 2011; pp. 2995–3004.
19. Ogata, M.; Sugiura, Y.; Osawa, H.; Imai, M. FlashTouch: Data Communication through Touchscreens. In Proceedings of the 31st Annual ACM Conference on Human Factors in Computing Systems (CHI 2013), Paris, France, 27 April–2 May 2013; pp. 2321–2324.
20. Kim, J.; Kasper, M.; Yeh, T.; Correll, N. SikuliBot: Automating Physical Interface Using Images. In Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST 2014), Honolulu, HI, USA, 5–8 October 2014; pp. 53–54.
21. Voelker, S.; Nakajima, K.; Thoresen, C.; Itoh, Y.; Øvergård, K.I.; Borchers, J. PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays. In Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS 2013), St. Andrews, Scotland, 6–9 October 2013; pp. 101–104.
22. Takashina, T.; Tamura, T.; Nakazumi, M.; Nomura, T.; Kokumai, Y. Toward a Compact Device to Interact with a Capacitive Touch Screen. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST 2016), Tokyo, Japan, 16–19 October 2016; pp. 173–174.
23. Kawahara, Y.; Hodges, S.; Cook, B.S.; Zhang, C.; Abowd, G.D. Instant Inkjet Circuits: Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (UbiComp 2013), Zurich, Switzerland, 8–12 September 2013; pp. 363–372.
24. Ikeda, K.; Tsukada, K. CapacitiveMarker: Novel Interaction Method Using Visual Marker Integrated with Conductive Pattern. In Proceedings of the 6th Augmented Human International Conference (AH 2015), Singapore, 9–11 March 2015; pp. 225–226.
25. Wiethoff, A.; Schneider, H.; Rohs, M.; Butz, A.; Greenberg, S. Sketch-a-TUI: Low Cost Prototyping of Tangible Interactions Using Cardboard and Conductive Ink. In Proceedings of the 6th International Conference on Tangible, Embedded, and Embodied Interaction (TEI 2012), Kingston, ON, Canada, 19–22 February 2012; pp. 309–312.
26. Yamanaka, S.; Kato, K.; Ta, T.D.; Tsubouchi, K.; Okuya, F.; Tsushio, K.; Kawahara, Y. SheetKey: Generating Touch Events by a Pattern Printed with Conductive Ink for User Authentication. In Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface (GI 2020), Toronto, ON, Canada, 21–22 May 2020.
27. Ikematsu, K.; Kato, K.; Kawahara, Y. LightTouch: Passive Gadgets for Extending Interactions on Capacitive Touchscreens by Automating Touch Inputs. In Proceedings of the Adjunct Publication of the 33rd Annual ACM Symposium on User Interface Software and Technology, Virtual Event, New York, NY, USA, 20–23 October 2020; pp. 10–12.

28. Okamoto, M.; Murao, K. Construction of a device that automatically generates various touch interactions. In Proceedings of the Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2019), London, UK, 9–13 September 2019; pp. 336–338.
29. Zimmerman, D.T.; Smith, R.J.; Paradiso, A.J.; Allport, D.; Gershenfeld, N. Applying electric field sensing to human-computer interfaces. In Proceedings of the 13th Annual ACM Conference on Human Factors in Computing Systems (CHI 1995), Denver, CO, USA, 7–11 May 1995; pp. 280–287.
30. Barret, G.; Omote, R. Projected-Capacitive Touch Technology. *Soc. Inf. Disp.* **2010**, *26*, 16–21. [[CrossRef](#)]
31. Foo, C.Y.; Lim, H.N.; Mahdi, M.A.; Wahid, M.H.; Huang, N.M. Three-Dimensional Printed Electrode and Its Novel Applications in Electronic Devices. *Sci. Rep.* **2018**, *8*, 7399. [[CrossRef](#)] [[PubMed](#)]