AroundSense: An Input Method for Gestures around a Smartphone

Kaho Kato[1], Kohei Matsumura[2], Yuta Sugiura[1]

[1]Keio University, 3-14-1, Hiyoshi, Kohoku, Yokohama, Kanagawa
[2]Ritsumeikan University, 1-1, Nojihigashi, Kusatsu, Shiga
Keywords: Mobile interaction, Hand gesture, Distance-measuring sensor

ABSTRACT

In this paper, we propose a gesture input method around a smartphone. Each gesture is detected by a distance-measuring sensor array attached to the side of a smartphone. We evaluated the accuracy of gesture recognition, and obtained an average accuracy of about 92.9% when identifying six distinct gestures.

1. INTRODUCTION

Smartphones are familiar things to us. They have many functions, such as Web browsing, taking photos, and so on, and we often use these functions.

We usually interact with our smartphones by touching the screen. However, it is sometimes difficult to interact with the screen. For example, finger gestures are not detected when our hands are dirty due to food or oil or when we are wearing gloves. In addition, gestures such as swipe and pinch sometimes hide displayed contents.

Many solutions to address these problems have already been proposed. Rafre [1] enabled users to interact by gestures detected via an internal proximity sensor on the upper screen while using a cooking application. The AQUOS PHONE ZETA SH-06E [2] can detect a fingertip above the screen and enables users to control the Web browser. By attaching sensors around a screen, the system can identify simple gestures and enables users to operate specific applications. They focused on the mid-air gestures above a smartphone, however, in this research, we focused on the gesture input around a smartphone located on a table. Matsumura proposed such various gesture interactions for multiple people [3]. Butler developed the gesture detecting system around a smartphone [4]. This system can detect the touched finger location. We aim to further increase the types of interactions that can be recognized.

In this paper, we focus on the area around a smartphone (Fig. 1). We propose an input method that uses various gestures for controlling a smartphone without interacting with the screen. Each gesture is detected by a distance-measuring sensor array attached to the side of a smartphone. The sensor data of each gesture is converted into two-dimensional (2D) grayscale image data. We use a support vector machine (SVM) for identification. We conducted an experiment in which participants performed a predetermined set of gestures and evaluated the accuracy of each gesture.

Fig. 1 Overview of the proposed input method

2. RELATED WORKS

2.1. Input around a Smartphone

Many input methods using the back of a smartphone have already been proposed. Matsushima et al. proposed an input method that uses image recognition to detect motions from a push button attached to the back of a smartphone [5]. Heo et al. attached several pressure sensors between a smartphone and its case to detect gestures by measuring the pressure levels at the back [6]. They combined some simple gestures to be detected by the pressure sensor and proposed multiple operations. Miyaki et al. also attached a pressure sensor to the back of a smartphone [7]. They combined some simple gestures to be detected by the pressure sensor and proposed multiple operations. Lv et al. developed a system that can detect the skin and outline of a hand in the real-time video of a smartphone camera and identify finger gestures [8]. The back of a smartphone is an effective input area when holding the smartphone. However, we focus to expand the interaction that can be used in situations where the phone is on the face.

Input methods around a smartphone have also been proposed. Hwang et al. developed various controllers that use magnets and a motion detection system that traces the magnetic fields of the controllers [9]. This system requires to use additional gadgets for controlling contents. In some situations, these gadgets can be cumbersome or unusable. Matsumura investigated user-defined gestures toward off-the-screen interactions [3]. He classified two types of gestures: used in common situations and dependent on contents.

To detect slide gesture around a smartphone, the methods using distance-measuring sensors are proposed. Butler developed the tap detecting system around a smartphone by the sensors attached to the sides of a smartphone [4]. The system is similar to what we are aiming for, but the user can use only a limited number of gestures. Kratz developed the gesture detecting system above the screen by distance-measuring sensors embedded in the side of a smartphone upward [10]. However, because the system requires the use of an area above the screen, contents may be hidden by user's finger during interaction.

In this research, we referred to the research of Matsumura [3] and Butler [4] et al., and focused on gesture input using the side of a smartphone. Our method does not obscure the screen, so users can concentrate on contents. Besides, we realize an input using various gestures.

2.2. Motion Detection by Photo Sensors

Many studies of motion detection by using various sensors have already been proposed. Yamashita et al. embedded photo reflective sensors in a head mounted display frame to measure the distance from the frame to users' cheek [11]. They implemented a recognition system that can be identified the cheek's shape changes when it was touched. Kikuchi et al. embedded photo reflective sensors into an earphone and measured the distance from different points on the earphone to the ear's skin [12]. They identified gestures from the skin changes in the ear and achieved "eyes-free" input. Miyata et al. developed a band-type device with embedded distance-measuring sensors and estimated hand posture grasping objects [13]. They detected the edge of a hand with respect to the grasped object and reconstructed the whole hand posture by using distance data measured by the photo sensors. Lim et al. embedded a distance-measuring sensor in the right side of a smartwatch and detected the position of the finger above the back of the hand [14]. They measured the distance between a sensor and a finger by sensing the intensity of an infrared light reflected on the finger. Nakatsuma et al. developed a watch-type device with embedded photo sensors and a piezoelectric sensor and developed an input system on the back of the hand [15].

In this research, by attaching distance-measuring sensors to the side of a smartphone, we measure the changes of the distance between sensors and fingers depended on gestures and estimate gestures.

2.3. Identification of Time Series Gestures

Kikui et al. applied machine learning for face recognition using a glass-type device embedded photo reflector sensors and decreased user's learning costs [16]. They made image data from the time series sensor data and learned by Convolutional Neural Network. Yoshinaga et al. developed a system that can detect "hand-snap" inputs by extracting histogram of oriented gradients (HOG) features from time series movie data taken by an RGB-D camera

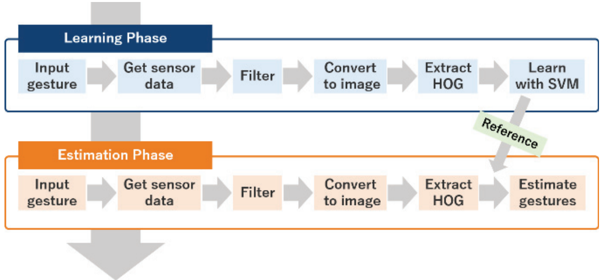

Fig. 2 Estimation methods of gesture

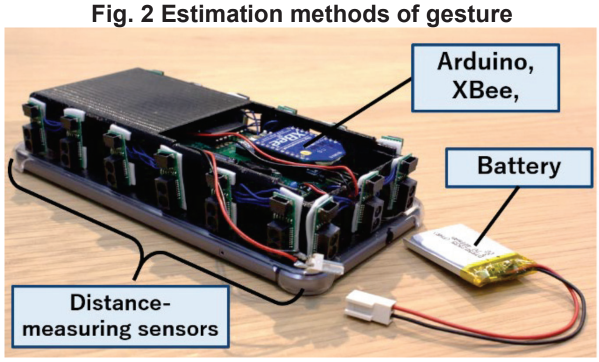

Fig. 3 Construction of the device

[17]. The system identifies hand shapes by using random forests for the HOG features. Fukui et al. developed a wristband-type device with embedded photo reflective sensors that can detect hand gestures by measuring changes in the wrist contour caused by hand gestures [18]. Obtained time series sensor data is converted into image

data and the system identifies by the HOG features and a SVM's classifier. Identification with HOG features is robust for deformation and position aberration.

Their method is effective to solve the gap of the input position or the gesture motion's starting timing. Our research estimates gestures by utilizing their method.

3. SYSTEM IMPLEMENTATION

3.1. Overview

In this research, we detect changes around a smartphone by using distance-measuring sensors. The sensors consist of an infrared LED and a position sensitive detector (PSD), which irradiates infrared light and receives light reflected from objects, respectively. The distance to the reflective point is calculated by the principle of triangulation based on an incident position of reflective light. Then, we can measure the distance from a sensor to an object's surface. When a user makes a gesture around a smartphone with his/her finger, we get as a sensor data a voltage data corresponding to the distance between each sensor and the finger. Obtained time series sensor data is converted into image data. We can detect and identify gestures by extracting the HOG features from them and a SVM's classifier (Fig. 2).

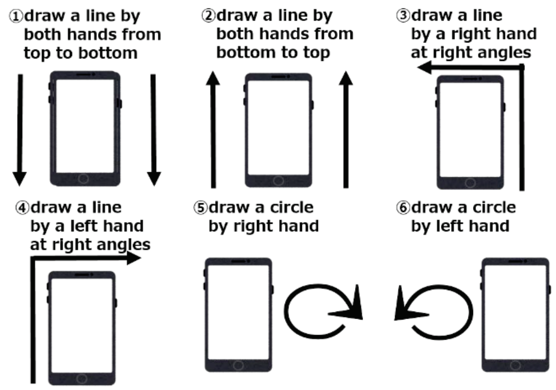

Fig. 4 Proposed gestures

3.2. Sensor Device

We embedded distance-measuring sensors into the side of a 3D-printed box that is attached to the back of a smartphone (Fig. 3). The distance-measuring sensors were GP2Y0E02A developed by Sharp Corporation, and the smartphone is a P10 lite developed by HUAWEI Corporation. The sensor can detect the distance from 4 [cm] to 50 [cm]. We attached 15 sensors to the smartphone. Two sets of seven sensors were positioned on both sides of the smartphone, and remainings were positioned on the top side. The box includes a microcontroller (Arduino Pro Mini 3.3V), wireless XBee module, and a Li-Po battery. Each sensor was connected to the microcontroller, and a sensor data was sent to the Arduino through the XBee module. We used Python 3.6 for implementation, and the system renewed every 25 [fps].

3.3. Gesture Design

We propose six gestures as input methods for a smartphone. These gestures are designed by referring to Matsumura's research [3]. The proposed gestures, shown in Fig. 4. These gestures are performed by using only the index finger of one hand or both hands.

3.4. Obtaining Learning Data and Identification

The system identifies the gesture by a SVM's classifier after we convert the time series sensor data into image data and obtain the HOG features. First, the system judges if a gesture has been performed. If the total value of all sensors passes a certain threshold, data collection starts. The system collects 20 frames of time series sensor data. The data is not stable when the distance to the objects is far, so we apply a median filter to get the median between recent six frames. Besides, we apply a low-pass filter (RC filter) to decrease a noise. The formula is as follows:

$$y[i] = a * x[i] + (1 - a) * y[i - 1]$$

(a: filter value, x: present data, y: time series data frame, i: frame number)

After obtaining the sensor data, we convert it into a 2D grayscale image data by means of normalization (Fig. 5).

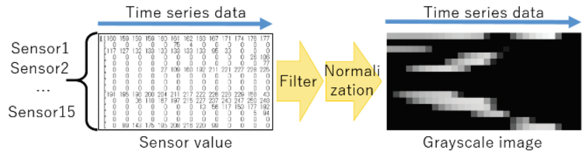

Fig. 5 Example of image data

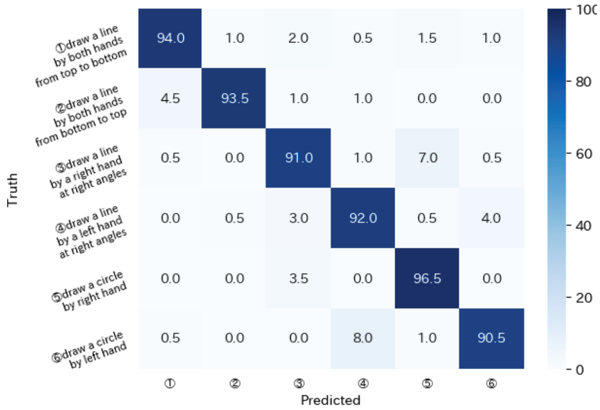

Fig. 6 Experimental results

In the converted image, each sensor value is allocated in the vertical axis direction and the time series data is allocated in the horizontal axis direction. We convert all sensor data into image data. We extract the HOG features from each image data and classify them by a SVM's classifier.

4. EVALUATION

4.1. Overview

In our experiment to determine the accuracy of our system, we recruited 10 participants (7 males, 3 females), aged between 22 and 25 years old (average: 23.4 years old). All of them were right handed. A smartphone used in the experiment was placed on a table.

Each participant performed all six gestures 20 times for a total of 120 gestures. The experimenter explained how to perform the gestures to them and they practiced the gestures a few times before any data was collected. As the sensor can detect from 4 [cm] to 50 [cm], we asked the participants to perform gestures at least 4 [cm] from the smartphone.

4.2. Result and Discussion

The total amount of data collected in the experiment was six gestures x 20 times x 10 person = 1200 gestures. The experimental results by leave-one-out cross validation are shown in Fig. 6. The system learned for each person, and the average accuracy of identification was about 92.9%.

Two gestures, "draw a line by both hands from top to bottom" and "draw a line by both hands from bottom to top", were sometimes misinterpreted as each other. These two gestures were performed in the same area, along both sides of smartphone. Our system starts to record sensor data when the sensor values change. If the recognition is inaccurate, an incomplete image data is created. So, the image data of these two gestures are similar. Such misinterpretation is not limited to only this combination. Other combinations, such as "draw a line by a right hand at right angles" and "draw a line by a left hand at right angles", and "draw a circle by right hand" and "draw a circle by left hand", are at a slight risk of being misinterpreted as each other.

5. LIMITATION AND FUTURE WORK

The distance-measuring sensor can be affected by other infrared light. In this system, the sensors irradiate infrared light and measure the distance between sensors and objects by using an angle of incidence of the reflected light. However, under sunlight conditions, the identification accuracy may decrease. For the future work, a filter system needs to be applied to prevent such a decrease in accuracy.

In this research, we proposed six gestures for smartphone input utilizing Matsumura's work [3]. However, we did not evaluate the usability of these gestures. For the future work, we will develop an application that can be controlled by the proposed gestures, and compare with the previous method.

6. CONCLUSIONS

In this paper, we proposed a gesture input method around a smartphone using distance-measuring sensors. The system does not obscure the screen, so it can provide a dynamic operation for a smartphone. The sensors were attached to the smartphone and measured the distance from a smartphone to a user's finger when they performed a gesture. Obtained time series sensor data is converted into image data. The system identifies gestures by the HOG features from the image data and a SVM's classifier. As a result of the evaluation, the system could identify gestures at the average accuracy of about 92.9%.

In the future, we will develop a real-time gesture identification system and make an application that uses the defined six gestures as input.

ACKNOWLEDGEMENTS

This work was supported by JST AIP-PRISM Grant Number JPMJCR18Y2.

REFERENCES

[1] KyoCera. “The guide How to use rafre”, https://www.kyocera.co.jp/prdct/telecom/consumer/lineup/kyv40/guide/index11.html, (accesses 2018-12-30).

[2] SHARP, “AQUOS PHONE ZETA SH-06E”, http://www.sharp.co.jp/products/sh06e/service04.html, (accessed 2018-12-30).

[3] K. Matsumura, “Studying User-Defined Gestures Toward Off the Screen Interactions”, Proc. ITS '15, pp.295-300 (2015).

[4] A. Butler, S. Izadi, and S. Hodges, “SideSight: Multi-"Touch" Interaction Around Small Devices”, Proc. UIST ’08, pp. 201–204 (2008).

[5] N. Matsushima, W. Yamada, and H. Manabe. “Attaching Objects to Smartphones Back Side for a Modular Interface”, Proc. UIST '17, pp.51-52 (2017).

[6] S. Heo and G. Lee, “Force gestures: augmented touch screen gestures using normal and tangential force”, Proc. CHI '11, pp.1909-1914 (2011).

[7] T. Miyaki and J. Rekimoto, “GraspZoom: zooming and scrolling control model for single-handed mobile interaction”, Proc. MobileHCI '09, Article 11 (2009).

[8] Z. Lv, A. Halawani, M. Khan, S. Réhman, and H. Li, “Finger in air: touch-less interaction on smartphone”, Proc. MUM '13, No.16, pp.1-4 (2013).

[9] S. Hwang, M. Ahn, and K. Wohn, “MagGetz: customizable passive tangible controllers on and around conventional mobile devices”, Proc. UIST '13, pp.411-416 (2013).

[10] S. Kratz, and M. Rohs, “HoverFlow: Expanding the Design Space of Around-device Interaction”, Proc. MobileHCI ’09, pp. 4:1–4:8 (2009)

[11] K. Yamashita, T. Kikuchi, K. Masai, M. Sugimoto, T. Bruce, and Y. Sugiura, “CheekInput: turning your cheek into an input surface by embedded optical sensors on a head-mounted display”, Proc. VRST '17, Article 19 (2017).

[12] T. Kikuchi, Y. Sugiura, K. Masai, M. Sugimoto, and T. Bruce, “EarTouch: turning the ear into an input surface”, Proc. MobileHCI '17, Article 27 (2017).

[13] N. Miyata, T. Honoki, Y. Maeda, Y. Endo, M. Tada, and Y. Sugiura, “Wrap & Sense: Grasp Capture by a Band Sensor”, Proc. UIST '16, pp.87-89 (2016).

[14] S. Lim, J. Shin, S. Kim, and J. Park, “Expansion of Smartwatch Touch Interface from Touchscreen to Around Device Interface Using Infrared Line Image Sensors”, Sensors, vol.15, no.7, pp.16642-16653 (2015).

[15] K. Nakatsuma, H. Shinoda, Y. Makino, K. Sato, and T. Maeno, “Touch interface on back of the hand”, Proc. SIGGRAPH '11, Article 19 (2011).

[16] K. Kikui, Y. Itoh, M. Yamada, Y. Sugiura, and M. Sugimoto, “Intra-/inter-user adaptation framework for wearable gesture sensing device”, Proc. ISWC '18, pp.21-24 (2018).

[17] T. Yoshinaga, T. Terada, M. Akaike, K. Furugori and M. Koga, “Hand-snap Gesture Recognition by Using Hand-motion and Hand-shape Estimated with Random Forest classifier”, Proc. CVIM ’13, No.19, pp.1-6 (2013).

[18] R. Fukui, S. Okishiba, H. Karasawa and A. Warisawa, “Dynamic Hand Motion Recognition Based on Wrist Contour Measurement for a Wearable Display”, Proc. Robomech ’17, vol.2017, pp.2A2-L02 (2017).