Development of Multiple Device Collaboration System Using Built-in Camera Image

Kazuki Tada(✉) and Jiro Tanaka

University of Tsukuba, Tsukuba, Japan
{kazuki,jiro}@iplab.cs.tsukuba.ac.jp

Abstract. In this paper, we introduce a multi-device collaboration system using the image obtained from the built-in camera. Users can use applications that utilize multiple devices in cooperation without needing special devices such as a touch panel sensor. Our system enables accurate position tracking of a smartphone any screen by obtaining an image from the device's front camera to recognize the device and measure the device's position by template matching [1]. In many multi-device collaboration approaches, it was necessary to overlap the screens. However, our proposed method is capable of multi-device collaboration without overlapping the screens of the devices, and it can be applied using a wide range of off-screens. We implemented some of the applications using this technique. In addition, we showed the usefulness of this approach by evaluation experiments.

Keywords: Multiple device · Image processing · Image recognition · Smartphone · Template matching

1 Introduction

Recently, due to the spread of smartphones and tablet devices, people have come to possess multiple devices. Multi-device collaboration systems permit users to connect many different devices and share the content among them. Such systems are beneficial in improving work efficiency.

Various techniques that recognize different devices collaborating with each other have been proposed. For example, Yatani et al. and Kamo et al. proposed methods to recognize other devices by capturing mobile devices equipped with a marker or infrared LED by room camera [1,2]. Furthermore, a method of overlaying the mobile devices with a conductive material on the touch screen device has been proposed [3]. In this method, since the touch point occurs on the touch screen, the positional relationship between collaborating devices can be determined.

However, in these techniques, special tools (e.g. a touch screen and infrared LED) are required for the multi-device collaboration. In this study, we have implemented a new multi-device collaboration method using a differential image obtained from the device's built-in camera. This approach uses only cameras mounted on the device to determine the positional relationship between multiple devices and implement various cooperative operations.

M. Kurosu (Ed.): HCI 2016, Part II, LNCS 9732, pp. 419–427, 2016.
DOI: 10.1007/978-3-319-39516-6_40

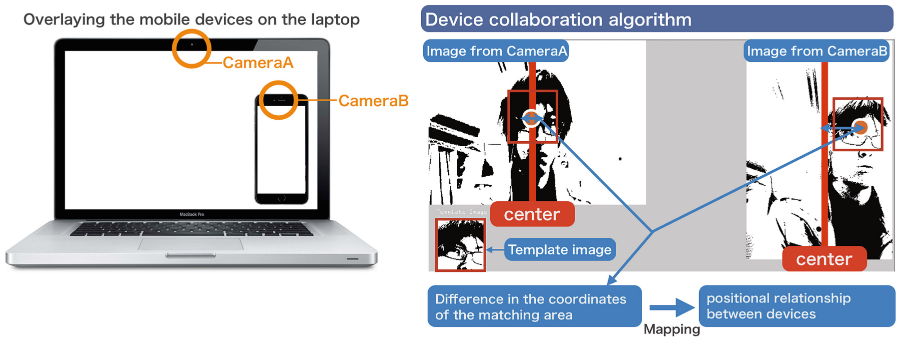

Fig. 1. Structure of device collaboration with built-in camera

2 Related Work

2.1 Multi-device Collaboration with Special Devices or Sensors

Multi-device collaboration methods have been proposed for a long time, many of which have used a sensor. Yatani et al. and Kamo et al. proposed methods to recognize other devices by capturing mobile devices equipped with a marker or infrared LED by room camera [1,2].

Swindells et al. [4] explored a device collaboration system using a digital pen. Cuypers et al. [5] proposed a method to determine the position of a mobile phone by detecting the flashlight that is mounted on it.

Liu et al. and Xu et al. have proposed a system to recognize the position information of the devices by using an acceleration sensor or magnetic sensor mounted on a smartphone and comparing the sensor value between the devices [6,7].

In these techniques, special tools (e.g. an acceleration sensor, magnetic sensor, and infrared LED) are required for multi-device collaboration. Moreover, these techniques cannot determine the position of the collaborating devices.

2.2 Multi-device Collaboration with Touch Screen

For the collaboration of the smartphone and surface, there are many proposed methods that utilize the characteristics of the touch screen. Hahne et al. [3] proposed a method of overlaying the mobile devices with a conductive material on the touch screen device. In this method, since the touch point occurs on the touch screen, the positional relationship between collaborating devices can be determined.

Strohmeier and Yasumoto et al. explored interaction scenarios based on touch events between devices [8,9]. They use the capacitive screen of a tablet computer to track the position of the phones touching it by using 'hand down' capacitance. This method was used by Chan et al. [10] for detecting tangible objects on displays.

2.3 Multi-device Collaboration with Device's Camera

Several multi-device collaboration methods using a camera have been proposed. Chan et al., Cuypers et al., and Rohs et al. proposed a method for device collaboration by reading a special pattern projected onto the display by the phone's camera [11–13].

THAW [14] is a method to overlap the mobile device to the computer screen. By analyzing the information obtained from the back camera of the mobile device, we can determine the positional relationship between collaborating devices. With a multi-device collaboration method using a camera, there is no need to attach extra equipment, such as additional sensors, to the smartphone.

However, it is necessary to project a special pattern on one of the displays or overlap the devices. In contrast, our proposed method is capable of collaboration without overlapping the screens of the devices, and it can be applied using a wide range of off-screens.

3 System Design

Our method uses an image obtained from the device's built-in camera to recognize the device and measure the device's position by template matching. For example, the collaboration between a personal computer and a smartphone is shown in Fig. 1 (left side). In this case, images of the same target are obtained with each of the cameras. A template is dynamically created to form part of the image of one camera. Next, our system tries to match this template image with images from the other camera. From the difference in the coordinates of the matching area, it calculates the positional relationship between the devices, as shown in Fig. 1 (right side).

3.1 System Flow

In this section, the operation of the system in this method will be described using a case where a user is linking a smartphone and laptop computer.

When an application that implements the proposed method is started on each device, the front camera becomes active to take the image. The image taken is sent to the server after the normalization and the binarization processing of the brightness. The server creates a template from the image transmitted from each device and attempts to match the image sent from the other devices. If the threshold of the match exceeds a prescribed value, it is determined that these two devices are in the coordination state (overlapping state), and calculates the difference between the matching position. Thereafter, the smartphone receives a signal from the server side. Then, the user performs calibration to adjust the relative position of the devices. Calibration is completed by performing a tap when the combined smartphone is in the lower right and upper left laptop. We show the system flow in Fig. 2.

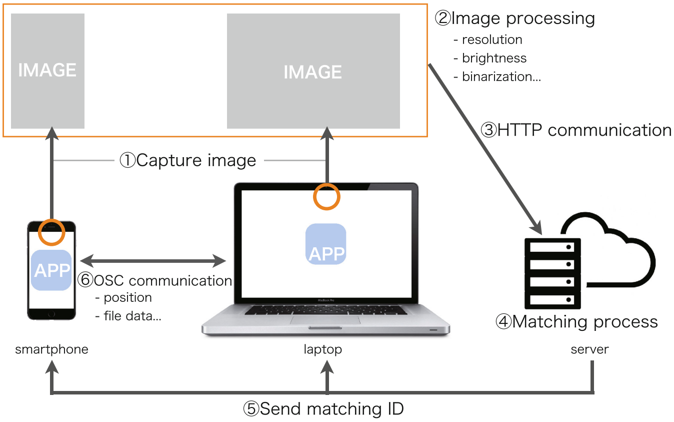

Fig. 2. System flow

4 Implementation

For the implementation, we used an iPhone 5 and 13-inch Macbook Air. The software was developed using C++ (with openFrameworks). For data communications between devices, we used HTTP for image processing for template matching and OSC (open sound control) for send device coordinates. In addition, we used the SSD (sum of squared difference) method as the template matching algorithm.

Hereinafter, by using images obtained from two cameras, we describe the methods to calculate the positional relationship between devices.

4.1 Normalization of Resolution

To correctly understand the positional relationship between the images taken from multiple devices, the resolution of each camera has to be constant (Fig. 3(a)). The adjustment of the resolution will be processed in accordance with the smallest side.

4.2 Normalization of Brightness

To correct the brightness of each camera, the normalized process luminance value has to be determined (Fig. 3(b)). The normalization is calculated based on the maximum brightness of the image obtained from a plurality of cameras. Furthermore, it is subjected to gamma correction, thereby suppressing the variation in brightness.

4.3 Binarization

To facilitate the template matching, the image obtained from the camera is binarized. The binarization uses the single manual threshold method and can adjust the threshold value to suit the environment in which the coordination is manually performed (Fig. 3(c)).

4.4 Making Template Image Dynamically

To generate the template image used for template matching, part of the video from one camera is extracted (Fig. 3(d)). The template image that shows the center of the object moving to the camera is automatically selected. For example, when the user is coordinating a laptop and smartphone, it is often the face of the person sitting in front of the laptop that is templated.

4.5 Matching Template Image with Other Images

Based on the template image generated using the previous technique, template matching of the image obtained from each camera is performed. When the matching is performed, the positional relationship between devices is calculated using the difference in the matching area coordinates of the image obtained from the camera (Fig. 1 left-side). The positional relationship is calculated by using the values of the calibration to be performed before collaboration and correction.

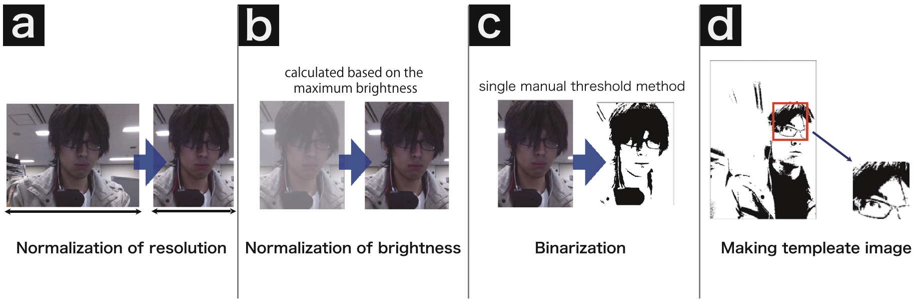

Fig. 3. Image processing algorithm

5 Applications

5.1 Translation Application

Figure 4 (left side) shows an example of using the translation application. When holding the smartphone over an English word on the computer screen, the translation appears on the smartphone. By using our method, without any special devices in addition to the collaborating devices, it is possible to implement these applications. By using such an application method, in addition to the translation application, it is also possible to make applications for, e.g. browsing for more detailed information.

5.2 Sub Display

Figure 4 (right side) shows an operational example of a screen sharing application using device collaboration with our method. In this application, it is possible to move the window on the screen of the personal computer to the smartphone. With conventional methods, the smartphone has to be superimposed on the screen of the personal computer. By using our method, without reducing the display area by covering the screen of the personal computer, it is easily possible to use another device as an extended display.

Fig. 4. (a) Example of translation (English to Japanese) application, (b) Example of screen sharing application.

6 Experiment

We implemented the techniques described in the previous section and performed experiments in the two environments (laptop-smartphone and tablet-smartphone).

6.1 Experiment Outline

We measured the recognition speed of the device and the deviation of the detected position. The devices used in the experiments are shown in Table 1. During this experiment, the brightness of the room was about 160 lux. In the experiments, the distance between the smartphone and laptop screen was set to zero.

Table 1. Device list

Device name	Width	Height
Macbook Air (13 in.)	325 mm	227 mm
iPhone 5	58.6 mm	123.8 mm
Iconia tab A500-10S	260 mm	177 mm

Table 2. Result of average recognition speed

	Macbook air	Iconia tab A500-10S
iPhone 5	0.5 s	0.6 s

Table 3. Result of average deviation of detected position

	Macbook air	Iconia tab A500-10S
iPhone 5	18.1 mm (90.5 px) and 14.6 mm (73 px)	10.4 mm (62.4 px) and 8.2 mm (49.2 px)

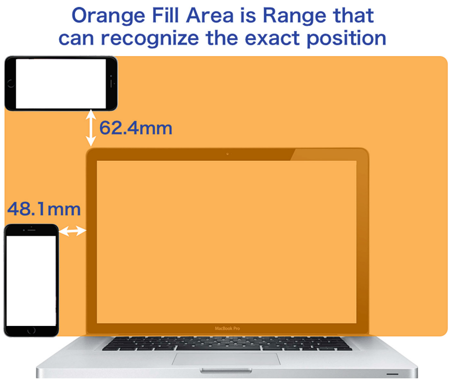

Fig. 5. Available range to measure device's exact position

Tables 2 and 3 show a summary of the combination of the devices and the measurement results. The experimental results revealed that it is possible to accurately perform collaboration, as shown in the next section.

In addition, our method can be used with a wide off-screen range, as shown in Fig. 5. When the device is within this range, it is possible to accurately measure the device's position (measurement error < 25 mm).

6.2 Considerations

The results of the experiment show that the tracking speed was at a practical level. However, the accuracy of the detected position has become inferior to that of previous studies. We think the cause is fluctuation in the brightness of the outside light. Therefore, we think these external factors should be compensated by the image processing.

This detection error is related to the size of the desktop icons. Therefore, our method can be sufficiently applied to applications requiring a rough location and orientation. For example, sub-display applications, such as those mentioned in Sect. 5.2, can be mentioned.

The most characteristic feature of this approach is that it can also cooperate in the off-screen. Therefore, we need to think of an example application that takes advantage of this feature.

7 Conclusion

In this paper, we showed a multi-device collaboration system using an image obtained from a built-in camera. This method is different from the existing approaches in that it does not require a special device for collaboration. We used the example of collaboration of a personal computer and smartphone and studied the collaboration accuracy. In future work, we are planning to improve the performance so that it can be adapted to big screens, such as table-top devices. Moreover, we are planning to perform a user study of applications using our technique.

References

1. Hiroki, K., Sugimoto, M., Yatani, K., Tamura, K., Hashizume, H.: Toss-it: intuitive information transfer techniques for mobile devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2005, pp. 1881–1884 (2005)
2. Kamo, H., Tanaka, J.: Interlocked surfaces: a dynamic multi-device collaboration system. In: The 15th International Conference on Human-Computer Interaction, HCII 2013, pp. 317–325 (2013)
3. Elstner, S., Hahne, U., Schild, J., Alexa, M.: Multi-touch focus+context sketch-based interaction. In: Proceedings of the 6th Eurographics Symposiumn Sketch-Based Interfaces and Modeling, SMIB 2009, pp. 77–83 (2009)
4. Dill, J.C., Swindells, C., Inkpen, K.M., Tory, M.: That one there! pointing to establish device identity. In: Proceedings of the 15th Annual ACM symposium on User Interface Software and Technology, UIST 2002, pp. 151–160 (2002)
5. Olwal, A., Lightsense: enabling spatially aware handheld interaction devices. In: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2006, pp. 119–122 (2006)
6. Sun, S.W., Cheng, W.H., Liu, K.W., Lin, I.P., Hsu, X.S.C.: G-spacing: a gyro sensor based relative 3d space positioning scheme. In: Proceeding of ACM SIGGRAPH 2015 Posters, SIGGRAPH 2015, Article No. 35 (2015)
7. Momeni, A., Xu, D., Brockmeyer, E.: Magpad: a near surface augmented reading system for physical paper and smartphone coupling. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology, UIST 2015, pp. 103–104 (2015)
8. Strohmeier, P.: Displaypointers- seamless cross-device interactions. In: Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology, ACE 2015, pp. 86–93 (2015)
9. Yasumoto, M., Teraoka, T.: Vistouch: dynamic three-dimensional connection between multiple mobile devices. In: Proceedings of the 6th Augmented Human International Conference, AH 2015, pp. 89–92 (2015)

10. Roudaut, A., Chan, L., Muller, S., Baudisch, P.: Capstones and zebrawidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, pp. 2189–2192 (2012)
11. Chan, L.W., Wu, H.T., Kao, H.S., Ko, J.C., Lin, H.R., Chen, M.Y., Hsu, J., Hung, Y.P.: Enabling beyond-surface interactions for interactive surface with an invisible projection. In: Proceedings of the 23th Annual ACM Symposium on User Interface Software and Technology, UIST 2010, pp. 263–272 (2010)
12. Vanaken, C., Reeth, F.V., Cuypers, T., Francken, Y., Bekaert, P.: Smartphone localization on interactive surfaces using the built-in camera. In: Proceedings of the IEEE International Workshop on Project-Camera Systems, Procam 2009, pp. 61–68 (2009)
13. Raubal, M., Essl, G., Rohs, M., Schoning, H., Kruger, A.: Map navigation with mobile devices: virtual versus physical movement with and without visual context. In: Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI 2007, pp. 146–153 (2007)
14. Heibeck, E., Maes, P., Leigh, S., Schoessler, P., Ishii, H.: Tangible interaction with see-through augmentation for smartphones on computer screens. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST 2014, pp. 55–56 (2014)