HoverLink: Joint Interactions using Hover Sensing Capability

Takuro Kuribara
University of Tsukuba
1-1-1 Tennodai, Ibaraki,
305-8573, Japan
kuribara@iplab.cs.tsukuba.ac.jp

Buntarou Shizuki
University of Tsukuba
1-1-1 Tennodai, Ibaraki,
305-8573, Japan
shizuki@cs.tsukuba.ac.jp

Jiro Tanaka
University of Tsukuba
1-1-1 Tennodai, Ibaraki,
305-8573, Japan
jiro@cs.tsukuba.ac.jp

Abstract

Users often connect two mobile devices at close range to transfer files such as pictures and movies from one device to another. In this paper, we present HoverLink, a new form of joint interactions on two mobile touchscreen devices such as smartphones and tablets *using hover sensing capability*. HoverLink allows users to connect two devices, manipulate the data on the devices, and disconnect them in a simple and continuous manner. We also describe HoverLink's applications: transferring data, and a menu interface.

Author Keywords

Mobile device; co-located collaboration; multi-device interaction; smart phone; file sharing; multi-surfaces; cross-device interactions.

ACM Classification Keywords

H.5.2. [User Interfaces]: Input devices and strategies, Interaction styles.

Introduction

Users often connect two mobile devices at close range to transfer files such as pictures and movies from one device to another (e.g., pictures on one's smartphone to his/her friend's tablet). While many technologies for transferring these data exist, such as Wi-Fi and Bluetooth, users need

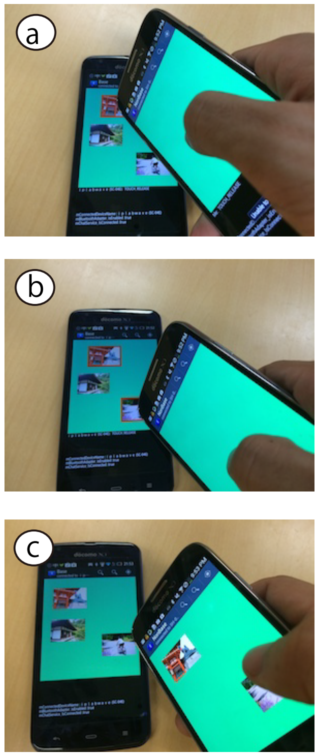

Figure 1: Joint interactions on two mobile touchscreen devices. a) The user starts the interactions by touching the base device's touch screen with a non-base device. b) The interactions are enabled. c) The user finishes the interactions by releasing the non-base device from the base device's touchscreen.

to select the devices to transfer to. For example, if a user wants to transfer some pictures from his/her mobile device to his/her friend's device through Bluetooth, they first make their devices scan other available devices, browse a list of the found devices, and then they select target devices among them.

Recent works to address the above problem have demonstrated various technologies to connect devices. For example, [5, 13] use accelerometers; [9, 10] use custom-made devices; and [4] uses sound.

In this paper, we present HoverLink, a new form of joint interactions [3] on two mobile touchscreen devices such as smartphones and tablets *using hover sensing capability*. Specifically, HoverLink allows users to connect two devices, manipulate the data on the devices (e.g., select data and transfer them), and disconnect them in a simple and continuous manner: users connect them by using one device (non-base device) to touch the touchscreen of another (base device) with hover sensing capability, manipulate data by using the non-base device, and disconnect them by releasing the non-base device from the base device.

The hover sensing capability is an emerging sensing technology which is already available on some smartphones such as AQUOS PHONE ZETA SH-06E, ELUGA P P-03E, and GALAXY S4 SC-04E. We found an interesting fact behind this capability that when a contact is made on a device (i.e., not fingers, hands, nor body) with a hover-enabled touchscreen, hover events, instead of touch events, occur on the contact points. By using this fact in combination with accelerometers, we realize a new rich form of joint interactions on mobile touchscreen devices without any additional custom-made device.

We also describe HoverLink's applications: transferring data, and a menu interface.

Related Work

Several researchers have proposed techniques for connecting multiple devices. Some techniques use synchronous actions such as bumping [5], tapping [11], pinching [8], or shaking [7] two devices together. SurfaceLink [4] allows users to connect multiple devices which are placed on a shared surface using swipe gesture between the devices. Chen et al. proposed a technique in which users wave a hand from one device to another to connect them [2]. Stitching [6] allows users to connect pen-operated mobile devices by dragging the pen from one device's display to another by crossing over the bezel. In HoverLink, users simply touch the base device's touch screen with a non-base device to select a device to connect. Moreover, HoverLink achieves joint interactions that use the coordinate of the contact point.

Schmidt et al. [12] and Alt et al. [1] proposed a technique to transfer data from a mobile device to a display, which uses a bump to detect the device contacted to the display. While they used a display with an IR touch frame or a camera to detect the device, we use a device with a hover-enabled touchscreen.

Some researchers also proposed techniques for transferring data between two devices. Yatani et al. proposed a toss or swing gesture to transfer data between devices [13]. FlashTouch [9] allows users to transfer data using a custom-made stylus. Pick-and-Drop [10] allows users to pick (select) data from a device and drop (transfer) them onto another device. In HoverLink, users select and transfer data on the base device by touching them with a non-base device.

HoverLink

We describe HoverLink and its example applications to explore the possibilities of HoverLink: transferring data, and a menu interface.

Description of Joint Interactions

Figure 1 shows joint interactions on two mobile touchscreen devices using HoverLink. First, users start the interactions by touching the base device's touchscreen with a non-base device as shown in Figure 1a. After this, users can manipulate data on the two devices using the non-base device as shown in Figure 1b. Finally, users finish the interactions by releasing the non-base device from the base device's touchscreen. Thus, users can easily connect two devices, manipulate the data on them, and disconnect them.

HoverLink uses a touchscreen device with hover sensing capability as a base device. We show how the hover sensing capability detects a finger and a device above and on the touchscreen in Figure 2. The capability senses not only a finger hovering above the touchscreen but also a device hovering above the touchscreen as shown in Figures 2a and 2c. When a contact is made between a finger and a touchscreen, touch events occur as casual touchscreen as shown in Figure 2b. However, interestingly, when a contact is made between a device and a touchscreen, hover events, instead of touch events, occur on the contact points as shown in Figure 2d.

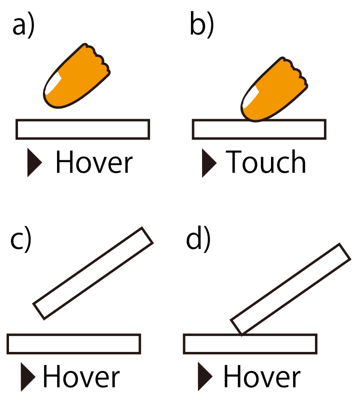

Figure 2: How the hover sensing capability detects a finger and a device above and on the touchscreen.

We use the above fact to implement HoverLink without any additional custom-made device. The system starts joint interactions when a hover event occurs on the base device's touchscreen and the accelerometers on both devices detect bumping at the same time; namely, the system distinguishes hover events by a finger and those by a non-base device with the accelerometer of the two devices. Note that hover events also occurred when other conductive materials (e.g., pen) hover above the touchscreen. Therefore, the system is designed to prevent these "false" hover events using the accelerometers. The system finishes the interactions when a hover event disappears from the base device.

Application: Transferring Data

HoverLink can be applied for transferring data between a base device and a non-based device in a smooth manner.

From a base to non-base device. Users can select a datum by touching it with the non-base device (Figure 3a). Users can select more data by dragging the non-base device on the base device's touchscreen. In this case, the data passed by the non-base device are selected (Figure 3b); passing already selected data undoes the selections. Users can select all the displayed data by tapping a non-base device's touchscreen while the two devices are connected (Figure 4a). After these selections, users can transfer the datum/data by flicking on the non-base device's touchscreen (Figures 3c and 4b). Users can cancel all the selections by releasing the non-base device from the base device's touchscreen.

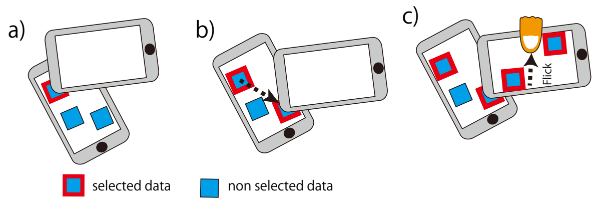

Figure 3: Selective transfer from a base to non-base device. Our interaction allows users to a) select a datum by touching, b) select more data by dragging, and c) transfer them by flicking on the non-base device's touchscreen.

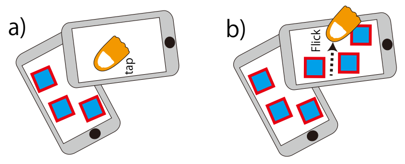

Figure 4: Full transfer from a base to non-base device. Our interaction allows users to a) select all the data by tapping a non-base devices's touchscreen while the two devices are connected and b) transfer them by flicking on the non-base device's touchscreen.

From a non-base to base device. Users can transfer data from a non-base device to a base device as shown in Figure 5. Users select some data by tapping the data on the non-base device while the two devices are connected (Figure 5a). After this selection, users can transfer them by flicking on the non-base device's touchscreen (Figure 5b).

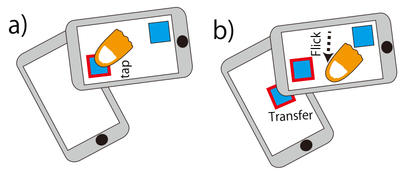

Figure 5: Selective transfer from a non-base to base device. Our interaction allows users to a) select some data by tapping the data on the non-base device and to b) transfer them by flicking on the non-base device's touchscreen.

Application: a Menu Interface
HoverLink is useful not only for transferring data, but also to use a non-base device as a menu interface. As an example, we present a paint application using HoverLink. In this application, we use a base device as a canvas and a non-base device as a tool palette and a pen (Figure 6a). The application allows users to draw a line on the canvas and change the line width simultaneously (Figures 6b and 6c). Note that the palette on the non-base device saves screen space on the base-device.

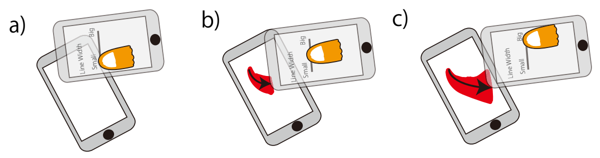

Figure 6: A menu interface using HoverLink. a) We use a base device as a canvas and a non-base device as a tool palette. b) For example, users can draw a line on the canvas and change the line width simultaneously, such as c) making the line thicker.

Feasibility

Since HoverLink depends on how well a hover event occurs in the base devices when the non-base devices contact on them, we conducted a study to evaluate how well a hover event occurs in the base devices when the non-base devices contact on them using various devices, as a feasibility of HoverLink. Note that, in a real world setting, users often protect their mobile devices with cases. Thus, this study uses both mobile devices with a case and ones without a case.

Apparatus
We used a Galaxy S4 SC-04E (Android 4.2.3) and an AQUOS PHONE ZETA SH-06E (Android 4.2.3) as the base devices. We also used twelve devices, which included the base devices, as the non-base devices as shown in Figure 7. We asked 9 volunteers (8 males and 1 female) aged 21-25 to use their mobile devices in this study; one volunteer had two devices (Figures 7i and 7j); two volunteers had the same model (Figures 7h Figure 7d). Two devices were our laboratory's ones (Figures 7a and 7b). Among the twelve devices, five were covered with a case and seven were not covered with a case.

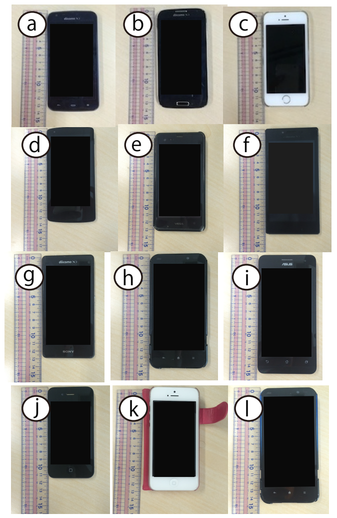

Figure 7: The list of the devices used in this study.

Procedure
Users (i.e., one author and the volunteers) were asked to contact a base device's touchscreen with their non-base device's edge, and then they were instructed to move/tilt the non-base device on the base device. The base device was placed on a flat surface. All events of the base device were recorded.

Results
Hover events occurred under all conditions as shown in Figure 8. However, both hover events and touch events occurred randomly in three devices under the condition that the users tilted the devices. Namely, hover events occurred in all devices (100%); hover events only occurred in nine devices (75%); touch events occurred in three devices (25%). In a comparison between the ones with a case and the ones without a case, the former produced only hover events (100%) and three of the latter produced touch events (43%). This might be because the three devices were not covered with a case and their bodies consist of conductive material; this electrically couples the users' hands with the touchscreen of the base devices through the bodies of non-base devices and thus touch events occurred.

ID	non-base device	case	event on SH-06E	event on SC-04E
a	SH-06E	none		hover
b	SC-04E	none	hover and touch	
c	iPhone 5s	used	hover	hover
d	Nexus 5	none	hover	hover
e	PTL21	used	hover	hover
f	L-01E	none	hover	hover
g	SO-02	none	hover and touch	hover and touch
h	SHL22	used	hover	hover
i	A500KL	none	hover	hover
j	iPhone 4	none	hover and touch	hover and touch
k	iPhone 5	used	hover	hover
l	SHL22	used	hover	hover

Figure 8: Results of the study.

In contrast, hover events occurred stably when the devices were covered with the cases. Therefore, HoverLink has the possibility to be used on various nowadays mobile touchscreen devices.

Implementation

We implemented the prototype of HoverLink as Android applications in JAVA.

We used Bluetooth to connect the devices. When users touch a base device's touchscreen with a non-base device, their Bluetooth is turned on. To detect this bumping, the accelerometers on both devices are used. Then, the base device scans for a non-base device whose Bluetooth is turned on and connects with it automatically. The devices are connected while hover events occur on the base device (this design involves one limitation: if users accidentally touch the base device's touchscreen, the connection is finished). The connection is finished when the users release the non-base device from the base device's touchscreen. And then, connected devices' Bluetooth is turned off.

Discussion and Future Work

In our current implementation, the base devices is connected to a non-base device whose Bluetooth is turned on. This means that if many devices' Bluetooth is turned on, unintentional connection may occur. Moreover, users must wait while the device's Bluetooth is turned on and scans for another device to connect. Therefore, we plan to adopt the same approach with another research (e.g., [5, 4, 1]): we connect all devices to the same network beforehand. In this case, all the remote devices (non-base devices) send its own time stamped accelerometer data to the server device (base device). The base device compares its own time stamped accelerometer data to the ones from non-base devices when a hover event occurs in the device. As another approach to avoid such unintentional connection, we will plan to use received signal strength indicator (RSSI). The RSSI is often used to estimate the distance between the devices. Therefore, the base device can select the closest device to connect with this approach.

When we tested our implementation, we observed that the coordinate of a hover event tends to be slightly different from the coordinate of the contacted point. This difference lowers accuracy in pointing during joint interactions. Thus, we plan to investigate the relationship between these points to adjust the difference to improve the usability.

Conclusion

We have presented HoverLink, a new form of joint interactions on two mobile touchscreen devices using hover sensing capability. The users can easily connect/disconnect two devices and manipulate the data without other custom-made devices. We also showed example applications of HoverLink. Our feasibility study demonstrated HoverLink can be used on various devices. In future, we plan to conduct user studies to measure the performance of HoverLink.

References

[1] Alt, F., Shirazi, A. S., Kubitza, T., and Schmidt, A. Interaction techniques for creating and exchanging content with public displays. In *CHI '13*, 1709–1718.

[2] Chen, K.-Y., Ashbrook, D., Goel, M., Lee, S.-H., and Patel, S. AirLink: Sharing files between multiple devices using in-air gestures. In *UbiComp '14*, 565–569.

[3] Chen, X. A., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: Exploring joint interactions on a smart phone and a smart watch. In *CHI '14*, 159–168.

[4] Goel, M., Lee, B., Islam Aumi, M. T., Patel, S., Borriello, G., Hibino, S., and Begole, B. SurfaceLink: Using inertial and acoustic sensing to enable multi-device interaction on a surface. In *CHI '14*, 1387–1396.

[5] Hinckley, K. Synchronous gestures for multiple persons and computers. In *UIST '03*, 149–158.

[6] Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: Pen gestures that span multiple displays. In *AVI '04*, 23–31.

[7] Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., and Gellersen, H.-W. Smart-Its Friends: A technique for users to easily establish connections between smart artefacts. In *UbiComp '01*, 116–122.

[8] Lucero, A., Keränen, J., and Korhonen, H. Collaborative use of mobile phones for brainstorming. In *MobileHCI '10*, 337–340.

[9] Ogata, M., Sugiura, Y., Osawa, H., and Imai, M. FlashTouch: Data communication through touchscreens. In *CHI '13*, 2321–2324.

[10] Rekimoto, J. Pick-and-Drop: A direct manipulation technique for multiple computer environments. In *UIST '97*, 31–39.

[11] Rekimoto, J. SyncTap: Synchronous user operation for spontaneous network connection. *Personal and Ubiquitous Computing 8*, 2 (May 2004), 126–134.

[12] Schmidt, D., Chehimi, F., Rukzio, E., and Gellersen, H. Phonetouch: A technique for direct phone interaction on surfaces. In *UIST '10*, 13–16.

[13] Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M., and Hashizume, H. Toss-It: Intuitive information transfer techniques for mobile devices using toss and swing actions. *IEICE Transactions on Systems and Computers E89-D*, 1 (2006), 150–157.