Memory Stones: An Intuitive Information Transfer Technique between Multi-touch Computers

Kaori Ikematsu
Ochanomizu University
g0920502@gmail.com

Itiro Siio
Ochanomizu University
siio@acm.org

ABSTRACT

Owners of multiple personal computing devices, such as mobile phones, tablet PCs, laptops, or desktop PCs, may frequently want to transfer information from one device to another. Whereas a drag-and-drop function on the same computing device is easy to achieve, it becomes cumbersome in an environment with multiple computing devices. We have to first locate and then select the target device from a list of devices on a network, even when the device is right in front of us. In this paper, a novel direct manipulation technique for executing drag-and-drop operations between multi-touch devices is proposed. Under our interface concept, dubbed "Memory Stones," a user can "pick up" a data object displayed on one device screen, "carry" it to another device screen, and "put it down" on that device using only their fingers. During this drag-and-drop operation, the user is invited to pantomime the act of carrying a tangible object (a "stone") while keeping their fingertip positions unchanged. The system identifies both the source and target devices by matching the shape of the polygon formed by the fingertips when touching each respective screen. We have developed a prototype system for small-to-large sized multi-touch computers including smartphones, tablet PCs, laptops, and desktop PCs, and have carried out a preliminary evaluation of its feasibility.

Keywords

Graphical User Interfaces (GUI); Drag-and-drop; Multi-touch Interaction; Gesture Input; User Interface for Multiple Computing Environment.

Categories and Subject Descriptors

H.5.2. [**Information Interfaces and Presentation (e.g. HCI)**]: Interaction Styles (e.g., commands, menus, forms, direct manipulation)

HotMobile'15, February 12–13, 2015, Santa Fe, New Mexico, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3391-7/15/02 ...$15.00.
http://dx.doi.org/10.1145/2699343.2699352.

1. INTRODUCTION

As computing devices become ubiquitous commodities in everyday life, the number of users who own multiple devices is sharply increasing, and it is no longer exceptional for a user to operate more than one computing device at the same time. Given this use scenario, there is an increasing need for an easy and intuitive user interface method for operations among multiple computing environments, such as drag-and-drop capability between two devices [14]. For example, if a user finds a dinner recipe on a Web page using a desktop or notebook PC, and wants to follow out the recipe in the kitchen while using a tablet PC, they have to find some way to drag-and-drop the URL of the Web page from the first device to the other. Similar needs arise frequently in everyday use: for example, presenting pictures or documents to friends or colleagues, using a nearby printer to make a hard copy of a document on a mobile device, or presenting a document on a laptop or tablet PC using a projection device in a meeting room. Such scenarios are already common in daily life and are likely to become more common in the near future. Although there are ways to transfer documents between two different computers, such as using a device-to-device network connection or a portable memory device, most of these methods include cloud file service involve cumbersome or complicated procedures and may be uncomfortable for average users. In this paper, we propose a new method we call "Memory Stones." This method involves the direct manipulation of multi-touch devices using finger gestures, metaphorically allowing the user to "pick up" a data object from one computing device, "carry" it to a nearby device, and "put it down" the data on that device.

2. RELATED WORK

Many methods have been developed for facilitating the transfer of information between multiple computers. In this section, we discuss previous work conducted in this area, which we have grouped into two categories: systems that allow a simple pairing between two computers before a conventional data transfer via a computer network, and systems that allow a seamless pairing and data transfer with a single operation.

2.1 Simple pairing

Several methods have been proposed for achieving an easy pairing between two computers before a conventional data transfer through a computer network takes place. For example, tranSticks [2] uses a pair of memory card devices and connects them in a one-to-one fashion by inserting them into

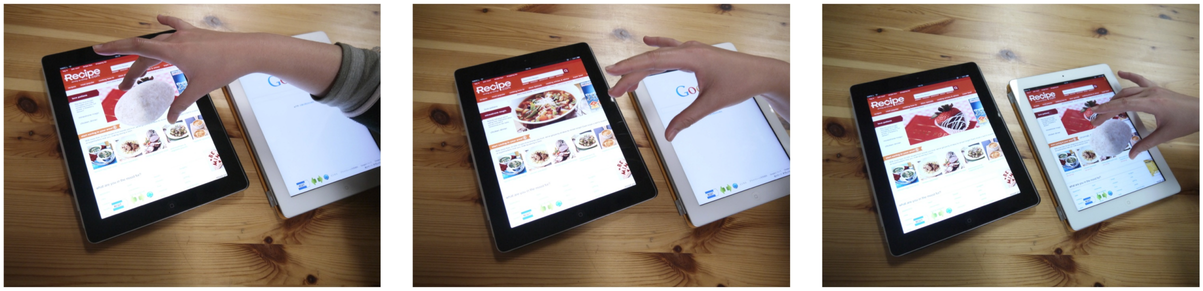

Figure 1: Proposed drag-and-drop method between two computing devices. A user picks up a data object (left) from one device using their fingers, (center) brings it to another device, and (right) puts it down on that device.

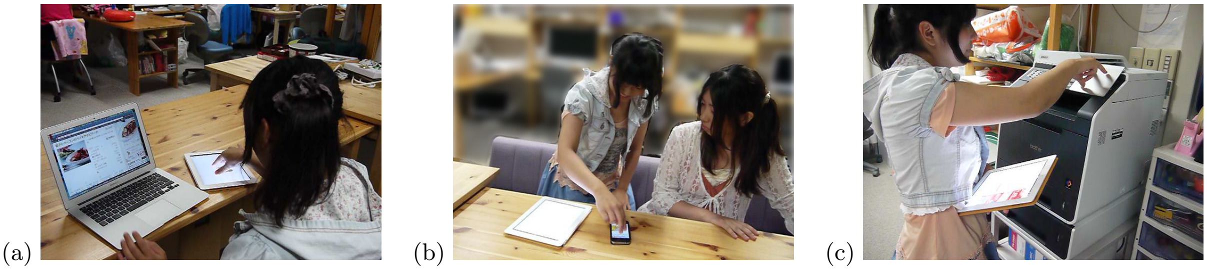

Figure 2: Examples of Copy-and-paste actions (a) from a laptop to a tablet PC, (b) from a smartphone to a tablet PC, and (c) from a tablet PC to a printer.

two different PCs. Using the accelerometer that is normally embedded in modern smartphones, Shake Well Before Use [10] establishes a network connection when the user holds two different mobile devices and shakes them both at the same time. In a similar way, Smart-Its Friends[8] and Synchronous Gestures [6] also use the accelerometer. To establish a connection, Smart-Its Friends uses a shaking gesture of two mobile devices held together, and Synchronous Gestures establish a connection when the user bumps two devices into each other. Point&Connect [13] uses a combination of the microphone and loudspeaker on a mobile phone, to allow the user to point their phone at a target phone with which they want to establish a link. Seeing-Is-Believing [11] uses a camera-equipped mobile device to identify a target device to which an optical ID marker is attached. Gaze-link [1] allows a use can establish a virtual connection over a network (e.g, Bluetooth or Ethernet) by looking at the target device through a device attached to a camera. SyncTap [15] links two computers together when the user taps or presses the input devices of both computers simultaneously. That one there! [17] uses infrared tags to identify paired computers. Finally, some commercial products and proposed methods facilitate computer pairing methods, including using NFC equipment or tags [1] , or detecting printers on the same network and listing them in a selection menu [2].

2.2 Seamless pairing and data transfer

A number of user-interface methods have also been proposed for achieving seamless drag-and-drop capability between two devices using implicit pairing functions. In Pick-and-Drop [14], using a pen device, a user can “pick up” a data object, such as an icon visible on a computer display, and “drop” it onto another computer display. This system uses an identifiable pen device to pair the two computers within a local area network. In Stitching [7], connection between two mobile devices is established by using a stroking gesture across both devices. This interaction technique requires the two devices to be contiguous. Therefore, a data transfer between two large desktop PCs or fixed displays is difficult to achieve. In Toss-It [19], an information transfer is achieved when a user makes a throwing gesture with the source device aiming for the target device. A background process identifies and spatially locates all computers within the shared physical space, and calculates a landing spot for the virtual object thrown by the user. The computer nearest to the landing spot will receive the virtual object. Hassan developed a similar technique that with a tilt-based interaction for sharing a document from a mobile device to a public display [5]. As a further examples of an information transfer technique in a multiple device environment, Wilson [18] uses multiple Kinect and projectors to enable users to establish connections through tracking gestures (e.g, swiping and grabbing). In addition, Andrew explored a meeting environment [3] for sharing and manipulating information across multiple mobile devices and public displays. Finally, using Touch & Interact [4], a user is able to share data between an NFC phone and a large display using a mesh of NFC tags.

Our proposed Memory Stones technique also aims to provide both information transfer capability and implicit pairing in a single, seamless operation. To develop a practical user-interface method, we restricted our technique to commonplace hardware devices or sensors that are available in consumer computing products. Therefore, we avoided meth-

[1] http://www.nfc-forum.org/resources/AppDocs/NFCForum_AD_BTSSP_1_0.pdf
[2] http://www.apple.com/iphone/features/airprint.html

ods that require infrared transceivers [17], cameras [18][3][9], or identifiable pen devices [14][12]. Moreover, because we also intended to support larger computing devices such as desktop PCs, laptops, and large-sized tablet PCs, acceleration-based methods [19][10], tilt-based method [5], and aim-based methods [13][11][17] were also avoided. Considering that multi-touch devices are becoming widespread in a wide variety of computers (smartphones, tablet PCss, laptops, and desktops) and are supported by many different OSs (iOS, Mac OS X, Android, Windows, Windows RT, and Windows Phone), we believe our proposed method has a practical application.

3. MEMORY STONES

In the real world, we can transfer objects such as printed documents by picking them up from one desktop and putting them down onto another. If the object we pick up is a solid material, such as a stone, the form of our fingertips will be unchanged over the course of such again. Our proposed Memory Stones method applies these common actions to a drag-and-drop operation between two devices equipped with multi-touch input. Using our method, a user can drag-and-drop an information object using the same actions they would normally apply to a physical object:"picking up" the information object from one device screen, "carrying" it to another device screen, and "putting it down" onto that screen. This is a more intuitive user interface method [16] than conventional methods that requiring a network connection processes, because it is based on more basic human activity than the use of a tool such as sticky pen [14] for bringing an object above a display surfac.

Figure 1 illustrates this sequence of user actions for copying a Web page. The user first picks up the Web page up by touching the screen with three fingers (Fig. 1(a)). Next, the user moves their hand to the target computer while keeping the fingertip positions unchanged, similar to carrying a solid object. (Fig. 1(b)). Finally, the user touches the target computer display using the same fingers and fingertip positions, at which point the Web page is copied to that computer (Fig. 1(c)) within few(2-3) seconds. Using this method, the user can drag-and-drop intangible information as if moving a tangible object. We have added visual feedback in the form of a virtual stone (a "Memory Stone") to both the source and target displays as an indication that the drag-and-drop action was accepted and completed. This visual feedback also reinforces the metaphor of carrying something tangible from one device to another.

Before and after the steps shown in Fig. 1 , the user can select the object to copy, and specify the location where it should be pasted, respectively. That is, after a user selects and starts to drag an object conventionally using a single finger, they can trigger the Memory Stone method by placing additional fingers onto the computer surface. Once the copied objects shown up on the target computer, the user can also trigger a conventional single-finger drag-and-drop operation by lifting all but one finger from the computer surface. This seamless transition between our method and a conventional drag-and-drop action enables a wide variety of operations, such as copying multiple icons and texts, and pasting them to specific locations on the target surface.

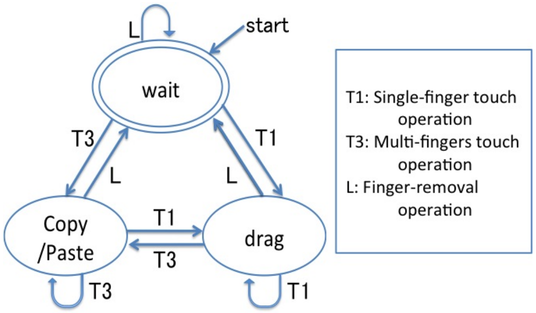
L
start
wait
T3
T1
L
L
Copy
/Paste
T1
T3
drag
T3
T1
T1: Single-finger touch operation
T3: Multi-fingers touch operation
L: Finger-removal operation

Figure 3: Deterministic finite automaton.

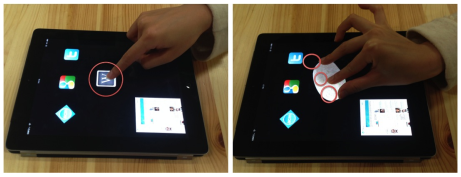

Figure 4: Object (icon) transfer method. After conventional dragging of an object (left), our method is activated by touching with multiple fingers (right).

Since our method is essentially an extension of a basic drag-and-drop operation to include nearby devices as paste destinations, it has many possible applications. Fig.2 shows some examples.

4. IMPLEMENTATION

To test the feasibility of our Memory Stones method, we implemented a prototype system in Objective-C for use on the iOS 5 and Mac OS X 10.7 operating systems. These platforms use a single programming framework to support multi-touch input on a wide range of devices, including smartphones (iPhones), tablet PCs (iPads), laptops (MacBooks), and desktop PCs (Macs and iMacs).

4.1 Client applications

One of our prototypes, shown in the Fig.1, functions as a simplified Web browser for computing devices running an iOS or OS X operating system. When a user touches a multi-touch display or trackpad of a device with multiple fingers, the URL of the Web page is copied to the system. The URL

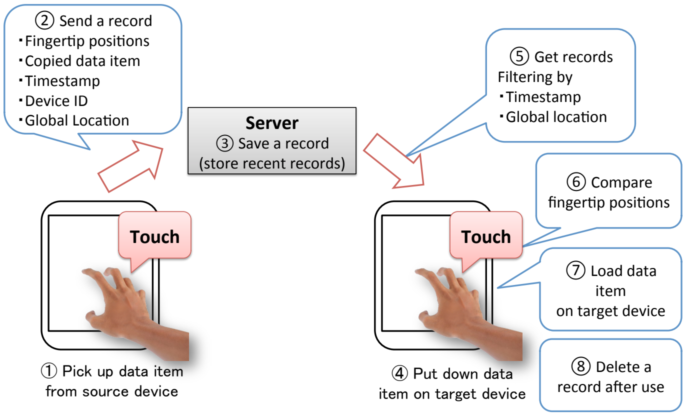

Figure 5: System configuration and operation.

is then pasted to the target device when the user touches the multi-touch display or trackpad of the target device with the same fingertip pattern, after which the pasted URL is loaded into the simplified Web browser running on the target device.

We developed other prototypes that support object selection before transferring data between devices. In these prototypes, a conventional drag-and-drop method is seamlessly combined before and after our data transfer method is applied. Figure 4 shows the method used to transfer selected objects. As shown on the left side of Fig.4, the conventional dragging of an icon is performed after it is selected. When the user places extra fingers on the touch surface (Fig. 4, right), a stone appeares and the inter-device drag-and-drop process is activated. When the user touches the destination device using the same finger shape, the stone image reappears (Fig. 4, right), and conventional dragging starts when the user removes all but her one finger (Fig. 4, left, in this case the user uses an index finger). The user can place the icon at the desired position using a conventional drag-and-drop method. As with a conventional text drag-and-drop method, the dragging of the characters starts when the user drags part of the selected text.When the user put extra fingers on the touch surface, a stone is appeared and our inter-device drag-and-drop process is activated. When the user touches the destination device by the same finger shape, the stone image also appears, and conventional dragging starts when the user lifts fingers other than one finger. The user can then place part of the text to the desired insertion point on the target device.

The transition state for these prototypes are shown in Fig. 3. T1 indicates a single-finger operation, T3 represents a multi-finger operation, and L refers to a finger-removal operation. When a user touches the computer surface with multiple fingers (T3), an inter-device drag or drop process starts. To indicate that the drag-and-drop operation was successful, the system displays the image of a stone on both the source and target device screen. The stone, which is scaled to fit within the polygon formed by the user's fingers, appears on the screen, and then disappears when the user lifts their fingers from the display (L). The same stone reappears when the user touches the destination device (T3) with a near-identical fingertip pattern, indicating a successful completion of the drag-and-drop operation.

Touching multiple fingers (T3) during a waiting state will activate the receiving or sending operations depending on the existence of valid data on the server. When no valid data is stored on the server, the application can select content(s) covered by the fingers and activate the transfer process to the destination device. This operation is adopted in a Web browser prototype (Fig. 1) for sending the whole page on the display.

In the current prototype, the user must apply a drag-and-drop operation using more than two fingers to avoid a conflict with other commonly adopted multi-touch two-finger operations, including pinching (zooming), rotating, sliding (scrolling), and tapping.

4.2 Server

Figure 5 shows the series of operations between devices when a user performs a drag-and-drop action. When a user touches the source computer to pick up an information object (e.g., Web page or image file), the client application pushes the following data to a prepared server: the timestamp, the global position of the computer, the URL of the data object, the unique device ID and the fingertip positions. The server preserves these data [3] .

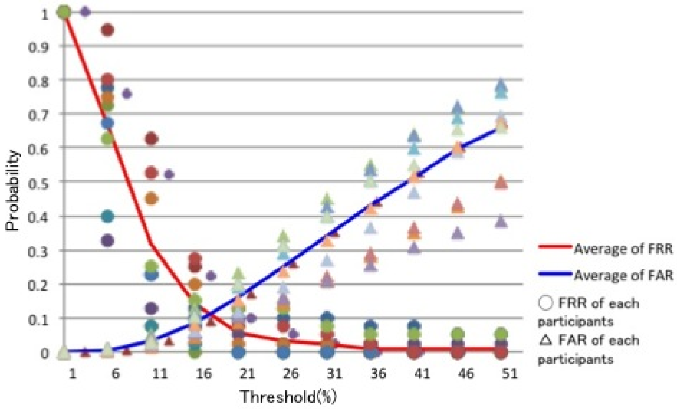

Figure 6: Matching threshold for the mean squared error of the distance between fingers and the average FAR and FRR values among the participants.

When the user touches the destination device to drop the information object, the client application on that device fetches the recent records [4] from the server. From these recent records received, those that are physically the farthest from the destination device (based on global positioning) are removed. After excluding unqualifying records, the program uses a polygonal congruence condition to try to find a record with a matching fingertip shape. Specifically, if the square summation of the difference between the lengths of the corresponding sides is less than a given threshold, then the two fingertip shapes are considered a match. If a match is found, the client application fetches the URL from the matched record and loads it into the browser, thereby completing the drag-and-drop procedure. In the current prototype, the URL fetch starts within 3.0 second after the user touches the destination device.

Global location information is used to prevent the establishment of an unintentional connection by accident. In addition to timestamp filtering, global location filtering prevents malfunction. We use the CoreLocation framework (supported by iOS and Mac OS, version10.6 or higher) to obtain the global location information. Windows, can also support Memory Stones since Windows PCs include Windows Sensor and Location (Windows 7) or a built-in position acquisition system (Windows 8). Our technique can be used even in non-GPS equipped PCs because the location acquisition served by the OS supports alternative positioning methods such as the use of the signal intensities of Wi-Fi base stations.

4.3 Threshold for Identification

Once the fingertip position data are acquired at the source and destination computers, the system matches the polygon formed by the fingertips as follows First, it calculates the distances between each pair of fingers, and sorts them by

[3] Five records are maintained in the current implementation.
[4] Finishes recording within 5 s in the current implementation.

length. Second, it calculates the mean squared error between each distance pair of the same order. If the polygons formed by the fingertips are exactly congruent, the mean squared error should be zero. Finally, in the third step, the system determines that the fingertip positions are identical if the mean squared error is less than the given threshold.

To determine an appropriate threshold, we asked ten female students (one graduate and nine undergraduate , nine right-handed and one left-handed) to perform a drag-and-drop action 40 times each (a total of 400 actions) between two tablet PCs (iPads), and obtained their fingertip position data. Using this data, we calculated the False Acceptance Rate (FAR) and False Rejection Rate (FRR), as shown in the following equations, to obtain several threshold candidates.

$$FAR = \frac{\text{Number of incorrect acceptances}}{\text{Number of trials}} \qquad (1)$$

$$FRR = \frac{\text{Number of incorrect rejections}}{\text{Number of trials}} \qquad (2)$$

FAR is the probability that the system incorrectly matches the fingertip position pattern to a non-matching pattern in the server, while FRR is the probability that the system fails to detect a match between the fingertip position pattern and a matching pattern in the server.

Figure 6 shows the average FAR and FRR values for all of the participants. The horizontal axis shows the threshold (% of length between fingers) for the mean squared error comparison. If the threshold is extremely low, the required matching accuracy is such that even a correct user's finger pattern may be rejected, and the FRR value is increased. On the other hand, if the threshold is very high, the required matching accuracy may be low enough to allow an incorrect user finger pattern to be accepted, and the FAR value is increased. We determined the threshold to be approximately 18% such that the value of FAR and FRR are equal; as a worst case, the Equal Error Rate (EER), which is the error rate at this point, is less than 11.0%. In addition, as we mentioned before, the actual system is expected to provide lower FAR value than the result of this experiment by using timestamp and location filtering. Furthermore we can expect an improvement in FRR when the user masters operation gestures. In that case, the system can provide a strict threshold option for skilled users to allow a more secure operation.

If a large number of people in a meeting room or classroom try to transfer data using the Memory Stones method, reliable data transfers may be difficult to achieve. One of our ideas to solve the problem is temporally place a stricter judgment on the fingertip-pattern similarity when a large number of requests arrive at the server at short intervals from the same location.

5. USER EVALUATION

In addition to the gesture performance experiment mentioned in the earlier threshold design section, we carried out two user-evaluation experiments. The first was to evaluate the mental effect of displaying a stone image when a user performs a drag-and-drop operation between devices. The other was to estimate the influence of an inclined touch surface, because in certain applications, such as those operating on a printer control panel (Fig.2) or public display, the input surface may not be placed horizontally. In these experiments, we asked five female graduate students (all right handed) to participate in the following experiments. All participants were familiar with smartphones, tablet PCs, and laptops with trackpads, and all had experience using multi-touch devices.

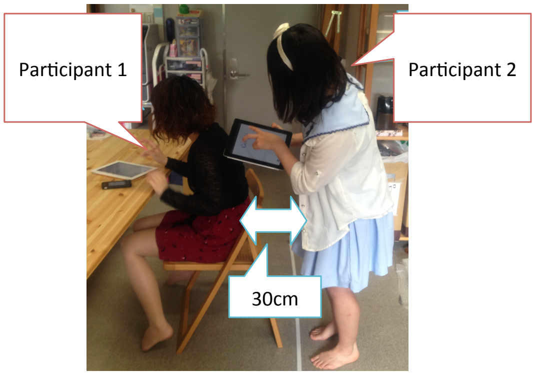

Figure 7: Setup of the shoulder surfing experiment.

5.1 Stone Image Effect

For the experiment on evaluating the stone image effect, we assigned each participant sixteen drag-and-drop tasks between two tablet PCs (iPads). Tasks with and without using stone image were conducted in turn. After the tasks were completed, we asked each participant the following question: "Of the two methods of interaction methods, (a) with the stone image and (b) without, which gives you a greater feeling of really transferring an object with your hand" The grading was based on a four-point scale: 4 = strongly agree, 3 = agree, 2 = disagree, and 1 = strongly disagree.

Each of the participants positively evaluated our method as intuitive and easy to use. The mean scores of the questionnaire results are 4.0 and 2.6 for with and without a stone image respectively. This shows that the task with a stone image is relatively more intuitive than without a stone image.

For the experiment on estimating the influence of an inclined touch surface, we assigned the participants data transfer tasks between an iPad placed on a table and another iPad tilted (90 or 45 degrees) relative to the horizontal surface. Each participant performed the task five times for both inclination angles (a total of 50 tasks), and we observed the influence of the operationality on the drag and drop operation based on the inclination angles of the devices. The success rate of the task for the 90-degree tilted iPad was 88.0% and for the 45-degree tilted iPad was 92.0%. Considering that the FRR is 10.0% for horizontally placed devices using the threshold mentioned in the previous section, the angle of the devices are not an obstacle, and it is therefore believed that Memory Stones can be applied to an inclined or wall-mounted display such as on a printer (Fig. 2) or a touch surface attached to a wall).

5.2 Shoulder surfing experiment

Although the proposed method facilitates secure drag-and-drop operations using temporal-personal fingertip positions, timestamps, and locations, there still remains the possibility that a malicious user can steal data by mimicking the fingertip action of an authorized user. One problem with the proposed application that information may be "stolen" by mimicking the formation of a user's fingertip positions. We therefore carried out a "shoulder surfing" experiment to estimate the possibility of data theft by an eavesdropper.

We asked five female undergraduate students (four right-handed and one left-handed) to participate in our experiment. We assigned them shoulder surfing tasks where they had to imitate the fingertip position of a user who was copy-and-pasting a data object using our proposed system. Figure 7 shows the experimental setup. Participant 2 executed shoulder surfing on iPad A 30-cm behind participant 1, who was sitting and copying-and-pasting data from a smartphone to iPad B, placed on a table, using three fingers. Participant 2 was allowed to move from side to side provided that she did not enter participant 1âĂŹs field of view. Each participant took the role of both participants 1 and 2, as shown in Fig. 9, and had 40 attempts at shoulder surfing. As a result, we recorded 200 trial datasets in this experiment, and the mimicked users were encouraged to simply "pick up a stone."

The results of this shoulder surfing experiment showed a success rate of about 4.0%. This value is less than one-half of the EER discussed in the previous section. Furthermore, because we employed both timestamp and location factors, as well as fingertip patterns, in the coupling process of the source and target computers, the person engaging in successful shoulder surfing had to conduct a false paste operation close to the target user within several seconds after the target user touched the transmitting device. Thus, we consider the threat of shoulder surfing to be acceptably low in practical situations.

6. CONCLUSION AND FUTURE WORK

We proposed a novel user-interface method called Memory Stones for executing drag-and-drop operations between nearby devices equipped with multi-touch inputs. We implemented a prototype of Memory Stones that runs on smartphones, tablet PCs, laptops, and desktop PCs, and carried out usability experiments. The participants of the usability experiments rated the proposed technique positively, and we found that users tend to move their fingers inward while performing drag-and-drop actions. The success rate of a shoulder surfing experiment was about 4.0%, and we therefore consider our method to be sufficiently secure for practical use.

We plan to enhance the scalability of our system by developing a more functional server-side application for managing, filtering, and searching the fingertip information, allowing the system to be used by larger numbers of users, and eventually becoming a worldwide service.

7. ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number 26330219.

8. REFERENCES

[1] Ayatsuka, Y., M. N., and Rekimoto, J. Gaze-link: A new metaphor of real-world oriented user interface. *IPSJ Journal 42*, 6 (2011), 1330–1337.(in Japanese).

[2] Ayatsuka, Y., and Rekimoto, J. transticks: physically manipulatable virtual connections. CHI '05, ACM (2005), 251–260.

[3] Bragdon, A., DeLine, R., Hinckley, K., and Morris, M. R. Code space: Touch + air gesture hybrid interactions for supporting developer meetings. ITS '11, ACM (2011), 212–221.

[4] Hardy, R., and Rukzio, E. Touch & interact: Touch-based interaction of mobile phones with displays. MobileHCI '08, ACM (2008), 245–254.

[5] Hassan, N., Rahman, M. M., Irani, P., and Graham, P. Chucking: A one-handed document sharing technique. INTERACT '09, Springer-Verlag (2009), 264–278.

[6] Hinckley, K. Synchronous gestures for multiple persons and computers. UIST '03, ACM (2003), 149–158.

[7] Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. Stitching: Pen gestures that span multiple displays. AVI '04, ACM (2004), 23–31.

[8] 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. UbiComp '01, Springer-Verlag (2001), 116–122.

[9] Lee, H., Jeong, H., Lee, J., Yeom, K.-W., Shin, H.-J., and Park, J.-H. Select-and-point: A novel interface for multi-device connection and control based on simple hand gestures. CHI EA '08, ACM (2008), 3357–3362.

[10] Mayrhofer, R., and Gellersen, H. Shake well before use: authentication based on accelerometer data. PERVASIVE'07, Springer-Verlag (2007), 144–161.

[11] McCune, J. M., Perrig, A., and Reiter, M. K. Seeing-is-believing: using camera phones for human-verifiable authentication. *Int. J. Secur. Netw. 4*, 1/2 (Feb. 2009), 43–56.

[12] Ogata, M., Sugiura, Y., Osawa, H., and Imai, M. Flashtouch: data communication through touchscreens. CHI '13, ACM (New York, NY, USA, 2013), 2321–2324.

[13] Peng, C., Shen, G., Zhang, Y., and Lu, S. Point&connect: intention-based device pairing for mobile phone users. MobiSys '09, ACM (2009), 137–150.

[14] Rekimoto, J. Pick-and-drop: a direct manipulation technique for multiple computer environments. UIST '97, ACM (New York, NY, USA, 1997), 31–39.

[15] Rekimoto, J. Synctap: synchronous user operation for spontaneous network connection. *Personal Ubiquitous Comput. 8*, 2 (May 2004), 126–134.

[16] Shneiderman, B. Designing the user interface 2nd edition.

[17] Swindells, C., Inkpen, K. M., Dill, J. C., and Tory, M. That one there! pointing to establish device identity. UIST '02, ACM (2002), 151–160.

[18] Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. UIST '10, ACM (2010), 273–282.

[19] Yatani, K., Tamura, K., Hiroki, K., Sugimoto, M., and Hashizume, H. Toss-it: intuitive information transfer techniques for mobile devices. CHI EA '05, ACM (2005), 1881–1884.