PAPER Special Section on New Technologies and their Applications of the Internet III

Toss-It: Intuitive Information Transfer Techniques for Mobile Devices Using Toss and Swing Actions

Koji YATANI[†a)], *Student Member*, Koiti TAMURA[†], Keiichi HIROKI[†], *Nonmembers*, Masanori SUGIMOTO[†], *and* Hiromichi HASHIZUME[††], *Members*

SUMMARY Mobile devices have acceleratedly penetrated into our daily lives. Though they were originally designed as a communication tool or for personal use, and due to the rapid availability of wireless network technologies, people have begun to use mobile devices for supporting collaborative work and learning. There is, however, a serious problem in mobile devices related to their user interfaces. In this paper, we try to alleviate the problem and propose intuitive techniques for information transfer, which is one of the typical usages of mutually-connected computers. Our system, Toss-It, enables a user to send information from the user's PDA to other electronic devices with a "toss" or "swing" action, like a user would toss a ball or deal cards to others. The implementation of Toss-It consists of three principle parts - gesture recognition, location recognition, and file transfer. We especially describe the details of gesture recognition and location recognition. We then evaluate the practicability and usability of Toss-It through the experiments. We also discuss user scenarios describing how Toss-It can support users' collaborative activities.
key words: *mobile devices, intuitive interfaces, information transfer, gesture recognition, location recognition*

1. Introduction

Mobile devices, such as cellular phones, PDAs (Personal Digital Assistants), have rapidly penetrated into our daily lives. These devices were originally designed as communication tools or for personal use. Due to the wide spread of wireless network technologies, people started to use mobile devices as support in collaborative work [1] and learning [2]. We believe that mobile devices will play a more important role for enhancing collaboration among people.

However, some of the serious problems in mobile devices related to their user interfaces, such as awkward input methods and small screens, have been pointed out. In order to solve these problems, a considerable number of researches have been conducted [3], [4]. Despite of these contributions, information transfer between mobile devices still forces users to do frastrating operations. Suppose you wanted to copy a file from your mobile device to others' mobile or electronic devices around you. Although a memory card or an infrared communication is usually available as information transfer methods, these approaches require several steps to complete a task, for example, (1) copy a file to a memory card, (2) move close to a person, (3) remove the card from your device, (4) insert the card into the other person's device, (5) copy the file to the person's device, and (6) remove the card and return it to your own device. When you want to pass a file to several colleagues, you have to conduct the same procedures repeatedly. In another case, if you wanted to print out a photo from your mobile device through a printer in front of you, you may have to know the information of the printer (e.g., its IP address, or its name on the network). Even if you know this information, you have to conduct frustrating operations on its graphical user interface, such as selecting several times menu items with a stylus pen, in order to specify the printer.

On the other hand, in the real world, when you pass something to another person around you, all you have to do is just tossing it toward the person. Our initial idea for this work is that if we could send information from our mobile devices to other devices as we would pass physical objects to others, we would be liberated from bothersome and awkward operations on our device. Therefore, we propose a system called Toss-It that enables users to transfer information in their mobile devices (PDAs in this work) in an intuitive manner by utilizing their mobility [5], [6]. Followings are examples of how Toss-It can be used:

- Pass a file from a user's PDA to another user's PDA with just a "toss" action toward him, or pass a file to another user's PDA beyond other users in-between with a stronger "toss" action as shown in Fig. 1 (a) (unicast transfer).
- Print out an image from a user's PDA through a printer with just a "toss" action toward the printer (Fig. 1 (b)).
- Project a slide onto a screen through a projector with just a "toss" action toward the screen.
- Pass a file from a user's PDA to several other users with just a "(horizontal) swing" action toward them as shown in Fig. 1 (c).

In order to allow for information transfer techniques with "toss" or "swing" actions, Toss-It must satisfy the following requirements:

Req1 Toss-It can recognize user's "toss" and "swing" actions conducted with his PDA.
Req2 Toss-It can automatically identify the positions and orientations of multiple users' PDAs and electronic devices.

Manuscript received March 11, 2005.
Manuscript revised June 13, 2005.
[†]The authors are with the University of Tokyo, Kashiwa-shi, 277–8561 Japan.
[††]The author is with National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: yatani@itl.t.u-tokyo.ac.jp
DOI: 10.1093/ietisy/e89–d.1.150

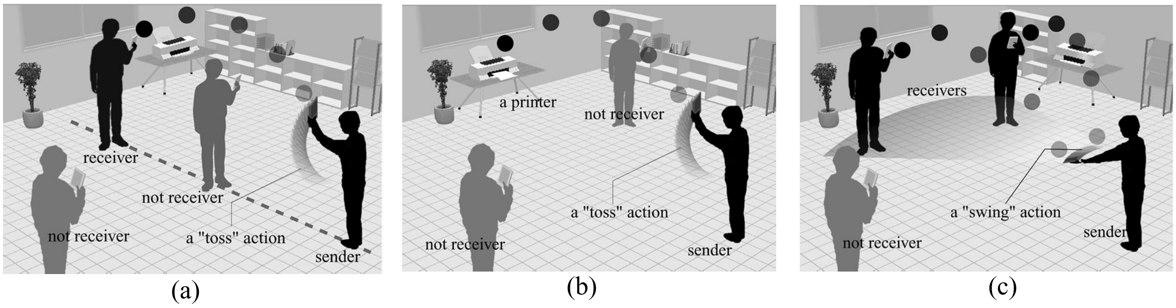


Fig. 1 Intuitive information transfer techniques with Toss-It. (a) From a PDA to another PDA, (b) from a PDA to a printer, (c) from a PDA to multiple PDAs.

Req3 Based on a user's action, Toss-It can transfer digital information from his PDA to other users' PDAs or to the corresponding electronic devices.

The final goal of our project is to satisfy the requirements without any external equipment embedded or installed in an environment where Toss-It is used, in order to make it available anywhere. Our initial goal, however, is to investigate if the proposed idea (information transfer by "toss" or "swing" actions) is possible and practical. This paper, therefore, discusses issues mainly related to Req1. In order to recognize users' actions, we attach inertial sensors to a PDA. This approach has several technical advantages. Especially it does not require complicated setting as compared to approaches such as using external equipment. For example, in a vision-based approach, installing and calibrating multiple cameras are necessary to capture a user's action in any location, which often bothers interface developers and designers.

As for Req2, there are several existing location recognition technologies applicable to Toss-It [7]. We are now developing a novel location and orientation recognition technology that can acquire relative positions and orientations of multiple devices without any external equipment, such as beacons in the ceiling. In the current implementation, however, we use a camera-based technology described later in this paper.

As for Req3, it may be possible to use P2P or ad-hoc network technologies. In this paper, however, we use a wireless LAN and a server computer. The server also manages data on positions and orientations of multiple devices. When a user conducts a "toss" or "swing" action with his PDA to transfer information, the software of Toss-It calculates the trajectory of the PDA, identifies target devices, and sends the information to the devices via wireless LAN.

This paper focuses on design and implementation issues for information transfer techniques among multiple mobile devices. The proposed techniques are applicable to various collaborative tasks. We will describe two usage scenarios for enhacing users' collaboration.

2. Related Works

Several research projects have proposed techniques for information transfer. In [8], a special stylus pen with memory enables a user to transfer information from one device to another in a "pick-and-drop" manner. [9] allows a user to conduct information transfer by pointing to the target device with RF pens and tags. In [10], a tangible user interface for intuitive information transfer techniques was introduced. However, these systems do not allow a user to send information to devices at a distance, nor to send it to multiple devices in an intuitive manner as shown in Fig. 1. Moreover, Toss-It allows a user to send information to a receiver beyond people in-between, because it sends information by a "toss" action, not by a "pointing" action.

There are some related works that utilize movements of a mobile device to provide more intuitive interfaces. For example, in [4], intuitive manipulation applications for mobile devices, such as maintaining viewing orientation when a device is rotated, or scrolling a screen by tilting a device, are proposed. XWand [11] is a universal remote controller. An accelerometer, a gyroscope, a magnetic sensor, and an image processing method are used for recognizing the location and orientation of XWand and identifying the users' actions with it. [4] shows two-handed interaction techniques for a PDA by combining pen input and a spatially aware display. Hinckley proposed a concept by using synchronous gestures for linking or manipulating multiple (mobile) devices [12], and several researches based on his concept have been conducted. SyncTap [13], for instance, links two devices by synchronously pushing the predefined buttons (e.g., an escape button) or tapping a screen of each device.

One difference between described systems and Toss-It is that they are used for recognizing relatively slow or small moves of mobile devices (e.g., tilting or rotating), but cannot be used for fast or large moves like "toss" or "swing" actions. Another important difference is that Toss-It allows users to link more than two devices simultaneously by one "swing" action. By utilizing a real-world metaphor (passing real objects to others), Toss-It can liberate users from bothersome manipulations on a graphical user interface, such as repeatedly selecting a computer from a list of icons.

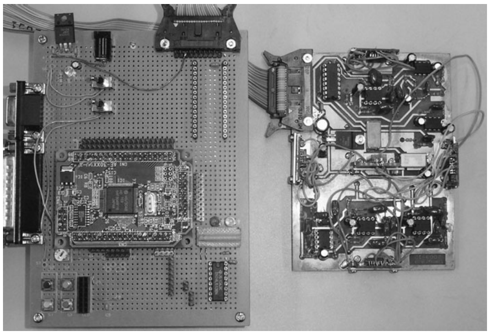


Fig. 2 A circuit board with inertial sensors.

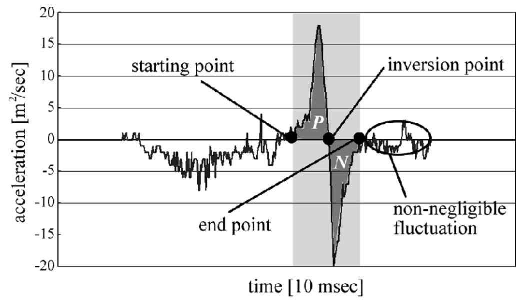


Fig. 3 A typical example of the output data of an accelerometer when a user conducted a "toss" action with Toss-It.

3. Gesture Recognition

3.1 Hardware

We have developed a circuit board that mounts accelerometers (Analog Devices ADXL210), gyroscopes (Murata ENC-03J, ENC-03M) and a microprocessor (Hitachi H8 microcomputer) as shown in Fig. 2. The circuit board is designed to be attached to a PDA. To capture users' quick actions as accurate as possible, four 2-axis accelerometers and three 1-axis gyroscopes are embedded in the board. They are connected to a PDA through a serial communication via a microprocessor.

3.2 Recognition Algorithm

In order to identify target devices that receive information by a user's "toss" or "swing" action, Toss-It is required to recognize not only the action, but also the strength of the "toss" action in a unicast situation (as shown in Fig. 1 (a)), and the trajectory of the "swing" action in a multicast situation (as shown in Fig. 1 (c)).

3.2.1 Eliciting "Toss" or "Swing" Actions

Ideally, Toss-It can recognize a "toss" or "swing" action through the output data of the inertial sensors. Several informal experiments, however, have indicated that non-negligible fluctuation occurs in the output data just after the action has been completed. Figure 3 shows a typical example of the output data of an accelerometer when a user conducted a "toss" action with Toss-It. In order to eliminate this fluctuation, we have devised a new recognition algorithm. To apply this algorithm, an assumption is made that a "toss" or "swing" action is initiated and finished in a state of rest. This assumption justifies the idea that the area of the positive part (P in Fig. 3) is equal to that of the negative part (N in Fig. 3).

The recognition process is summarized as follows: First, Toss-It searches an intersecting point of the output data curve and the zero acceleration line as shown in Fig. 3. When Toss-It has found a new intersecting point, it calculates the integral of the acceleration values between the intersecting point and the previous intersecting point, named the "starting point". If the value is greater than a specified threshold, Toss-It regards the current intersecting point as the "inversion point" and begins to calculate the integral of the acceleration values from the inversion point. While Toss-It makes the calculation, it evaluates the summation of the two integral values (the integral between the starting and inversion points, and the integral from the inversion point). When the value of the summation becomes approximately zero, Toss-It stops the calculation and regards the current point as the "end point". Finally, Toss-It recognizes that a user's action happened between the starting point and the end point (the highlighted region in Fig. 3).

In order to calculate the strength of a "toss" or the trajectory of a "swing", a transformation matrix between the absolute coordinate system and the PDA coordinate system must be determined. During a user's action, Toss-It updates an Euler matrix by using angular velocities gained through the gyroscopes, and calculates the transformation matrix.

3.2.2 Estimating the Strength of a "Toss" Action

Toss-It estimates the strength of a "toss", in order to determine how far "tossed" information travels and which devices receive the information. After several informal experiments of a "toss" action, we have made an assumption for reasonably accurate estimations and less complex calculations: When we toss something, we release it at the maximum speed. A "toss" action is started at the vertically downward position to the floor and finished without a follow through. Toss-It regards a point of the maximum velocity during the "toss" action as a release point of "tossed" information. Toss-It also calculates the launch angle by integrating the data from the gyroscopes. After determining the maximum velocity and the launch angle, Toss-It estimates the flying distance with the equation of motion.

3.2.3 Estimating the Trajectory of a "Swing" Action

Toss-It calculates the trajectory of a user's "swing" action through the second integral of accelerations. To determine how many degrees a user has swung his PDA around him, we assume that a trajectory of a "swing" is an arc. Toss-It calculates an angle of the arc, and recognizes the devices inside the arc angle as receivers of information as shown in Fig. 1 (c). To simplify the calculation of the angle, we use only horizontal moves and neglect virtual moves of a "swing" action. The radius of the arc is decided by each user's arm length. An arc angle of a user's "swing" action can be calculated with the chord length and the radius of an arc.

4. Location Recognition

For position and orientation recognition of a user's PDA, we use a marker with infrared (IR) LEDs and a stereo camera. Figure 4 shows the marker used in our method. Three IR LEDs are arranged to form an isosceles triangle (A, B and C in Fig. 4). For identification, a different blinking pattern for each user's PDA is assigned to the IR LED at the point A.

The recognition algorithm captures the shape of the isosceles triangle formed by the LEDs mounted on a user's PDA, and identifies the blinking pattern through the stereo camera. It considers the center of mass of A, B, and C as the user's PDA position and the vector sum of $\vec{BA}$ and $\vec{CA}$ as the user's PDA orientation.

5. Experiments and Evaluations

5.1 Experiments for Gesture Recognition

Preliminary tests proved that Toss-It could correctly distinguish between a "toss" (vertical move) and a "swing" (horizontal move). We, therefore, evaluated Toss-It with respect to how accurately it could recognize receivers of "toss" and "swing" actions. Six subjects participated in the following two experiments. In the first experiment, the subjects were asked to conduct a "toss" action and send information to devices placed at three different locations (1 [m], 2 [m], and 3 [m] away from a subject). In the second experiment, the subjects were asked to conduct a three different horizontal "swing" with their PDAs (45 [deg], 90 [deg], and 135 [deg]). Each subject repeated "toss" and "swing" actions 25 times for each of the three locations and three angles, respectively.

Figure 5 shows the distribution of the estimated distances of "toss" actions. Figure 6 shows the distribution of estimated angles of "swing" actions. The average and standard deviation (SD) for each target distance and those for each target angle are described in Table 1 and in Table 2, respectively. Differences of "toss" actions between two target distances (1 [m] and 2 [m], or 2 [m] and 3 [m]) and those of "swing" actions between two target angles (45 [deg] and 90 [deg], or 90 [deg] and 135 [deg]) proved to be statistically significant by a Welch's t-test (two-tailed, $p < .01$), respectively.

5.2 Experiments for Location Recognition

We conducted an experiment for evaluating the location recognition method. In the experiment, we recorded the estimated values of the position and orientation of a marker placed at the distance of 1.5 [m] from the camera, and calculated the measurement errors. Because the stereo cameras are installed on the ceiling looking down vertically onto the floor, orientations parallel to the floor are important. Therefore, when measuring orientations, we set the marker on a plane parallel to the floor. We then determined how accu-

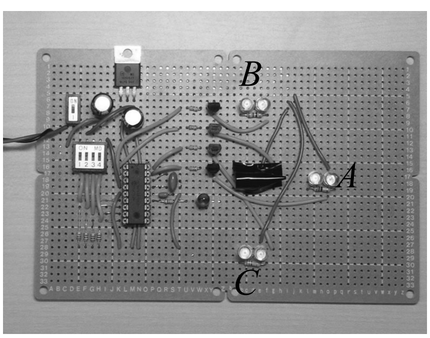


Fig. 4 A marker with infrared LEDs.

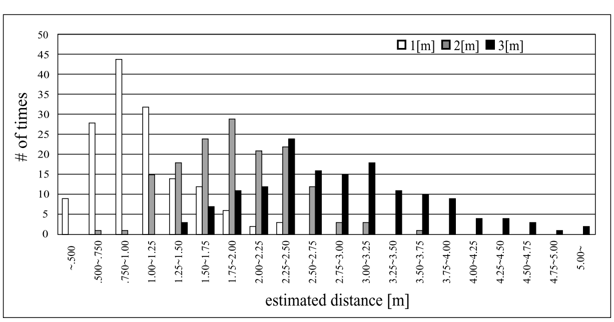


Fig. 5 A distribution of the estimated distances.

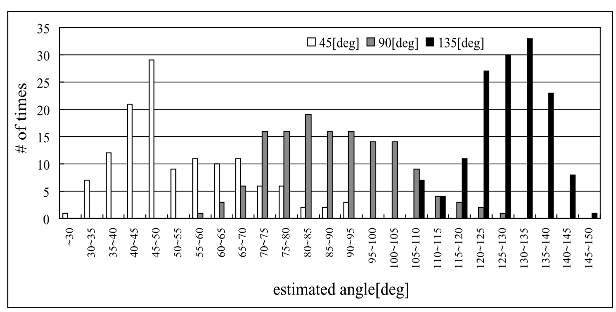


Fig. 6 A distribution of the estimated angles.

Table 1 Average and standard deviation for each target distance.

Target distance [m]	Average [m]	SD
1	1.06	0.418
2	1.90	0.513
3	2.88	0.806

Table 2 Average and standard deviation for each target angle.

Target angle [deg]	Average [deg]	SD
45	54.2	14.8
90	89.2	14.3
135	129	8.80

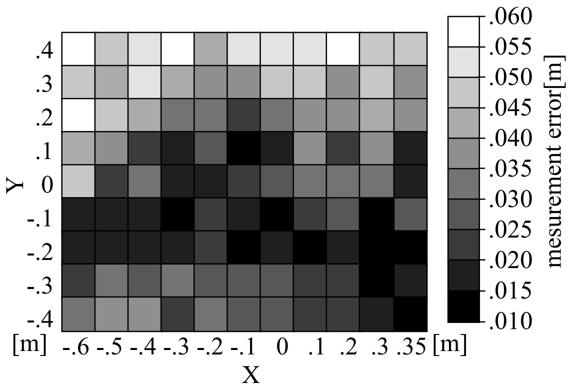


Fig. 7 The error distribution in the location estimation.

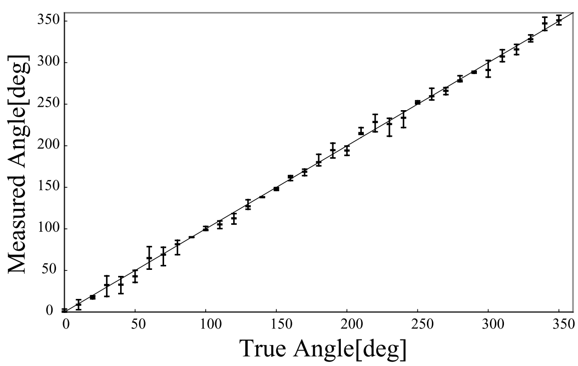


Fig. 8 The error distribution in the orientation estimation (at the edge of the camera image).

rately the markers were identified.

Figure 7 shows the error distribution in the position estimation. As shown in Fig. 7, the measurement errors increase when a marker is placed away from the center of the camera. The maximum value of the measurement errors is approximately six [cm] at the edge of the camera image.

Figure 8 shows the error distribution in the orientation estimation at the edge of the camera image. As shown in Fig. 8, even though the variation of the measurement errors is maximized at the edge of the camera image, it is about 10 [deg].

We confirmed whether the location recognition method can distinguish five kinds of the blinking patterns for the identification. As the result of the experiment, we observed that the location recognition method can distinguish the blinking patterns in less than two [sec]. When a user conducted a rapid movement, such as a "toss" or "swing" action, the location recognition method could not acquire the blinking pattern. However, when a user stayed still after the movement, it could acquire the blinking pattern and identify the user's PDA again.

5.3 User Studies

Figure 9 shows the experimental setting for evaluating Toss-It. Five subjects (male, right-handed) participated in the user studies, and four of them were asked to conduct the following tasks: E1 and E2 to transfer information by "toss" actions to (user1) and (user2), respectively, and E3 and E4 to transfer information by "swing" actions to (user1, user2, user4) and (user1, user2, user3, user4), respectively. The subjects were asked to conduct each of the four tasks ten times. In the experiments for "toss" actions, Toss-It transferred information to the nearest person within a two-meter radius from a landing point estimated through a sender's "toss" action (If no person existed in the area, no information transfer was done.). The results are summarized as Table 3. In the studies, each trial of the tasks by the subjects was judged as "success", if all and only the target users received the information. (e.g. in E6, user1, user2, and user4). The analyses of the user studies clarified the following issues:

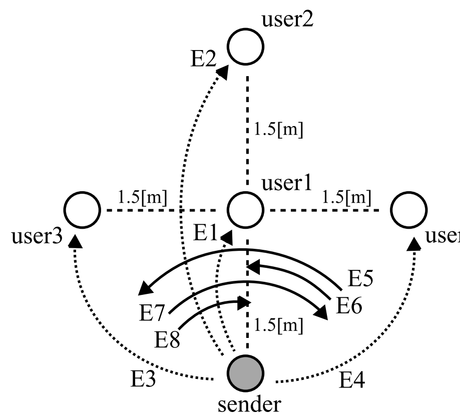


Fig. 9 Users' positions and four experiments in the user study.

Table 3 Success rate of information transfer in the experiments.

Exp.	Success Rate [%]	Exp.	Success Rate [%]
E1	75.0	E5	85.0
E2	75.0	E6	60.0
E3	80.0	E7	90.0
E4	70.0	E8	55.0

- Distance estimation errors: In 10.0 [%] of all the trials in E1, Toss-It sent information to user2, because estimated distances of "toss" actions were between 2.25 [m] and 5 [m]. In 2.5 [%] of all the trials in E2, estimated distances by Toss-It were between 5 [m] and 8 [m], and about 1 [m] in 20.0 [%] of the trials.
- Orientation recognition errors: In 10.0 [%] of all the trials in E1, Toss-It sent information to user4. This means that the recognition error of a sender's orientation was more than 20 degrees (about a half of the angle formed by the lines from the sender to user1 and user4), which is much larger than the maximum orientation recognition error (less than 10 degrees). Video analyses of the user studies clarified that although the sender conducted "toss" actions by exactly facing toward the target receiver, his PDA did not always direct exactly toward the receiver. This type of failure also happened in the trials in E2, E3, and E4.

- Results in "swing" actions toward all users: When the sender conducted "swing" actions toward all users in E5 and E7, the angle of the performed "swing" actions resulted to be wide than the expected 90 degree range. Therefore, estimated angles were also big enough, why the success rate of E3 was higher than other experiments. Subjects also commented, "To pass information to all, I just "swing" widely, so it's very easy."
- Results in small "swing" actions: When the sender tried to conduct small "swing" actions in E6 and E8, also here the actions were wider than originally foreseen. This caused more unsuccessful information transfer. In the experiments, we observed that one subject "cocked" his PDA before swinging to throw receivers more successfully. To be more precise, one subject first set his PDA right beside his body, then started to "swing" toward the target receivers.

6. Discussions

6.1 Issues to Improve Success Rates

Our experiments suggest that more precise estimation of users' "toss" and "swing" actions are necessary. We are reexamining the current design of Toss-It with respect to hardware and software aspects. In the current version of Toss-It, the sampling rate of the output from the inertial sensors is set to 10 [msec], due to the limitation of the microprocessor's capacity. This sampling rate, however, has turned out to be often insufficient for precisely recognizing the users' actions, especially rapid or quick actions. We are currently developing a revised version of the hardware with a different microprocessor that allows a higher sampling rate.

It may be effective to improve the transfer success rate by capturing all of the users' gestures or utilizing contextual information. For instance, suppose that receivers express their intention to receive information by tilting their own PDA vertically. Toss-It first identifies candidates of the receivers conducting the tilting gesture, and then sends information to some of them determined by sender's "toss" or "swing" actions. It may also be possible to utilize orientations of users' PDAs. For example, when their PDAs do not face toward a sender, Toss-It judges that they have no intention to receive information from the sender. Capturing users' gestures or orientations is also effective for blocking the users to receive unnecessary information.

By utilizing situational and contextual information, the transfer success rate will be improved. For example, it is reasonable that Toss-It regards user2 as the correct receiver in E2, when the estimated location is much beyond the user2's location. If Toss-It does not restrict candidates of receivers to those within a two-meter radius from a landing point calculated through a sender's "toss" action, the transfer success rate of E2 increases to 72.5 [%].

6.2 Other Issues to Be Examined

To make Toss-It more practical in a realistic setting, the following problems must be addressed in the future:

- Users of devices that Toss-It has judged as receivers do not want to receive information.
- While a user conducts a "toss" or "swing" action for information transfer, other users intentionally or unintentionally come too close to a calculated destination point or area, and therefore, the users receive or intercept the information.

To solve these problems, a certain confirmation process (e.g. accept/reject to send/receive information) among users would be required. However, from an HCI (human-computer interaction) point of view, it is desirable that this process should not detract Toss-It from its intuitiveness of the user interface. There will be several ways to implement this confirmation process. One idea is to develop the following mechanism as the "secure" mode:

- When Toss-It has determined the receiver devices through the calculation of a user's "toss" or "swing" action, it notifies users of the receiver devices in an intuitive manner, for example, giving users tactile feedback by vibrating motors attached to the devices.
- A popup window that includes a description of a user profile of a sender device, and types of information (e.g., text, image) and buttons ("receive" or "reject") appears on each receiver device. A user of the receiver device touches the "receive" button, if he/she wants to receive the information, otherwise touches the "reject" button.
- A popup window that includes a description of a device (or user) where the "receive" button was selected, and buttons ("send" or "reject") appears on the sender device. The user of the sender device touches the "send" button, if he/she wants to send the information to the device, otherwise touches the "reject" button.

The "secure" mode as described above is presented from the receivers' point of view. The "secure" mode from the viewpoint of senders is described as follows:

- When Toss-It has determined receiver devices through the calculation of a user's "toss" or "swing" action, a popup window appears on the sender's device. The pop-up window displays a description of a device (or a user) which Toss-It regards as the receiver devices, in addition to confirmation buttons ("send" or "dismiss"). The user of the sender device touches the "send" button, if he/she wants to send the information to the receiver devices, otherwise touches the "dismiss" button.
- Toss-It then lets the users of the receiver devices know that the information transfer request has arrived from the sender by providing them with tactile feedback.
- A popup window on the receiver's device displays a

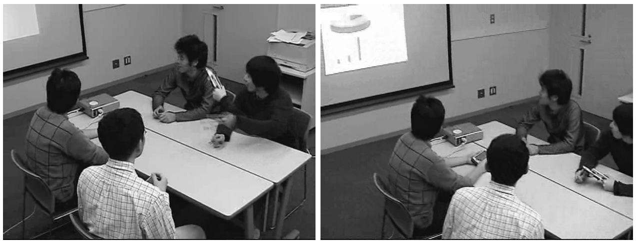


Fig. 10 A user presents his slide onto the screen by his "toss" action.

Fig. 11 User1 passes his material for the meeting to others by his "swing" action.

description of a user profile of a sender device, and the types of information (e.g., text, image) and buttons ("receive" or "reject"). A user of each receiver device touches the "receive" button, if he/she wants to receive the information, otherwise touches the "reject" button.

Through these confirmation processes, Toss-It can avoid cases where a user sends important information to unintended people or receives unnecessary information from unknown people. On the other hand, in the "usual" mode, this process is not required for information transfer. Users can select either of the modes based on what type of information they want to transfer.

The current version of Toss-It can be improved by integrating the solutions described above. We believe that through the improvements, the proposed information transfer techniques will become more practical and usable.

7. User Scenarios

Toss-It user scenarios could look as follows: When people start a discussion, one participant can make his slide in his PDA appear on the screen by his "toss" action, as shown in Fig. 10. The users are then ready to start their presentation or their brainstorming, because he is not required to conduct bothersome or time-consuming tasks, such as connecting a cable between the projector and his mobile device. In another case, a user simultaneously passes his presentation material in his PDA to others by his "swing" action, as shown in Fig. 11. Then he can immediately share necessary information for the meeting. As seen in the two examples, Toss-It can support not only information transfer between mobile devices but also users' collaboration such as during a meeting with their mobile devices, in an easy and intuitive manner.

8. Conclusions and Future Works

In this paper, we described the current implementation of Toss-It, especially the methods of gesture recognition and location recognition. We conducted experiments to evaluate each recognition method and user studies to evaluate how practical the proposed techniques are. We then discussed about how Toss-It should be improved.

Several issues still remain to be investigated. We plan to conduct intensive usability studies in order to evaluate Toss-It as a user interface for mobile devices. We will also explore possibilities for various applications based on the Toss-It architecture, such as a universal remote controller (for manipulating multiple devices simultaneously by a "swing" action, or a device behind an obstacle with a "toss" action), and entertainment games.

Acknowledgements

We appreciate students in University of Tokyo, who participate in our experiments and user studies.

References

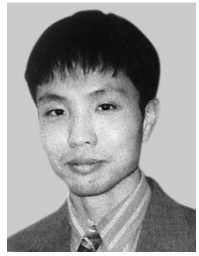
[1] M. Wiberg, "Roamware: An integrated architecture for seamless interaction in between mobile meetings," Proc. International ACM SIGGROUP Conference on Supporting Group Work, pp.288–297, 2001.
[2] E. Soloway, C. Norris, P. Blumenfeld, B. Fishman, J. Krajcik, and R. Marx, "Log on education: Handheld devices are ready-at-hand," Commun. ACM, vol.44, no.6, pp.15–20, 2001.
[3] I.S. MacKenzie and R.W. Soukoreff, "Text entry for mobile computing: Models and methods, theory and practice," Human-Computer Interaction, vol.17, no.8, pp.147–198, Aug. 2002.
[4] K.P. Yee, "Peephole displays: Pen interaction on spatially aware handheld computers," Proc. SIGCHI Conference on Human Factors in Computing Systems, pp.1–8, 2003.
[5] K. Yatani, K. Tamura, M. Sugimoto, and H. Hashizume, "Information transfer techniques for mobile devices by toss and swing actions," Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp.144–151, 2004.
[6] K. Yatani, K. Tamura, K. Hiroki, M. Sugimoto, and H. Hashizume, "Toss-it: Intuitive information transfer techniques for mobile devices," Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems, pp.1881–1884, 2005.
[7] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," Computer, vol.34, no.8, pp.57–66, Aug. 2001.
[8] J. Rekimoto, "Pick-and-drop: A direct manipulation technique for multiple computer environments," Proc. Annual ACM Symposium on User Interface Software and Technology, pp.31–39, 1997.
[9] C. Swindells, K.M. Inkpen, J.C. Dill, and M. Tory, "That one there! pointing to establish device identity," Proc. Annual ACM Symposium on User Interface Software and Technology, pp.151–160, 2002.
[10] B. Ullmer, H. Ishii, and D. Glas, "mediablocks: Physical containers, transports, and controls for online media," Proc. Annual Conference on Computer Graphics and Interactive Techniques, pp.379–386, 1998.
[11] A. Wilson and S. Shafer, "XWand: UI for intelligent spaces," Proc. SIGCHI Conference on Human Factors in Computing Systems, pp.545–552, 2003.
[12] K. Hinckley, "Synchronous gestures for multiple persons and computers," Proc. Annual ACM Symposium on User Interface Software

and Technology, pp.149–158, 2003.
[13] J. Rekimoto, "SyncTap: Synchronous user operation for spontaneous network connection," Personal and Ubiquitous Computing, vol.8, no.2, pp.126–134, 2004.

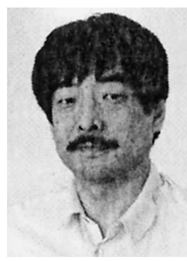
Koji Yatani received B.E. and M.S. from University of Tokyo in 2003 and 2005, respectively. He is currently a Ph.D. candidate in the Department of Frontier Informatics, Graduate School of Frontier Sciences at University of Tokyo. He is also a research fellow of the Japan Society for the Promotion of Science (DC1) since April 2005. His research interests are human-computer interaction, computer-supported collaborative work, entertainment computing and cognitive science. He is also a student member of ACM, IEEE and IPSJ.

Koiti Tamura received B.E. from the Faculty of Engineering at University of Tokyo in 2004, and enrolled in a master's course in the Department of Frontier Informatics, Graduate School of Frontier Sciences at University of Tokyo. His research interests are human-computer interaction, mobile computing, real-world computing and information retrieval.

Keiichi Hiroki received B.E. from the Faculty of Engineering at University of Tokyo in 2004, and enrolled in a master's course in the Department of Frontier Informatics, Graduate School of Frontier Sciences at University of Tokyo. His research interests are human-computer interaction and mobile computing.



Masanori Sugimoto received B.Eng., M.Eng., and Dr.Eng. degrees from University of Tokyo, Japan, in 1990, 1992, and 1995, respectively. In 1995, he joined Research and Department Development, National Center for Science Information Systems (currently National Institute of Informatics), Tokyo, Japan. He became an associate professor in Information Technology Center, University of Tokyo, in August 1999. Currently, he is an associate professor in Department of Frontier Informatics, Graduate School of Frontier Sciences, University of Tokyo. He was a visiting scientist in Department of Computer Science, University of Colorado, Boulder CO, USA (from January 1997 to November 1997). His research concern is related to human-computer interaction, especially mixed reality, mobile computing, computer supported collaborative work/learning, intelligent systems, and so on. Dr. Sugimoto is a member of IEEE, ACM, ISLS, JSAI, IPSJ, and JCSS.



Hiromichi Hashizume received B.E., M.E. and D.E. degrees in Electronic Engineering from University of Tokyo in 1979, 1981 and 1984 respectively. He is currently a professor of information media research, National Institute of Informatics, Japan. His research field was computer networks and telecommunication, however, recently his interest has shifted to mathematical modeling of communication systems.