Information Transfer Techniques for Mobile Devices by "Toss" and "Swing" Actions

Koji Yatani Koiti Tamura Masanori Sugimoto
Graduate School of Frontier Sciences
University of Tokyo
{yatani, tamura, sugi}@itl.t.u-tokyo.ac.jp

Hiromichi Hashizume
Multimedia Information Research Division
National Institute of Informatics
has@nii.ac.jp

Abstract

In recent years, mobile devices have rapidly penetrated into our daily lives. Several drawbacks of mobile devices have been mentioned so far, such as their limited computational capability, small screen real estate, and, so on, as compared with notebook or desktop computers. However, by fully utilizing the most notable feature of mobile devices, that is, mobility, we can explore possibilities for a new user interface of the devices. In this paper, we use PDAs and propose intuitive information transfer techniques for them, which could not be achieved with notebook or desktop computers. By using the system called *Toss-It,* a user can send information from the user's PDA to other electronic devices with a "toss" or "swing" action, as the user would toss a ball or deals cards to others. We have developed a circuit board designed to be attached to a PDA and algorithms for recognizing "toss" and "swing" actions. Preliminary user studies of ***Toss-It*** indicated that it could correctly identify receivers of information by "toss" or "swing" actions. Our research project is in progress, and this paper describes the current status of the project by focusing on issues related to HCI (Human Computer Interaction). We will discuss about several critical issues to be investigated in our future studies.

1. Introduction

In recent years, mobile devices have rapidly penetrated into our daily lives. Several drawbacks of mobile devices have been mentioned so far. such as their limited computational capability, or small screen real estate, and so on, as compared with notebook or desktop computers [13]. However, by fully utilizing the most notable feature of mobile devices, that is, mobility, we can explore possibilities for a new user interface of the devices.

Suppose you want to copy a file in your mobile device to others' mobile devices or electronic devices around you. Although a memory card or an infrared communication is available as information transfer methods, these methods 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 person's device, (5) copy the file to the person's device, and **(6)** remove the card and return it to your device. When you want to pass a file to several colleagues, you have to conduct the same procedures repeatedly. In another case, if you want to print out a photo in your mobile device through a printer in front of you, you have to conduct frustrating operations on its graphical user interface, such as selecting menu items with a stylus pen several times, in order to specify the printer.

On the other hand, when you pass something to another person around you in the real world, all you have to do is just tossing it toward the person. You can also pass something to multiple people simultaneously as you would deal cards. If you can send information from your mobile device to other devices as you would pass physical objects to others, you will be liberated from bothersome and awkward operations on your device.

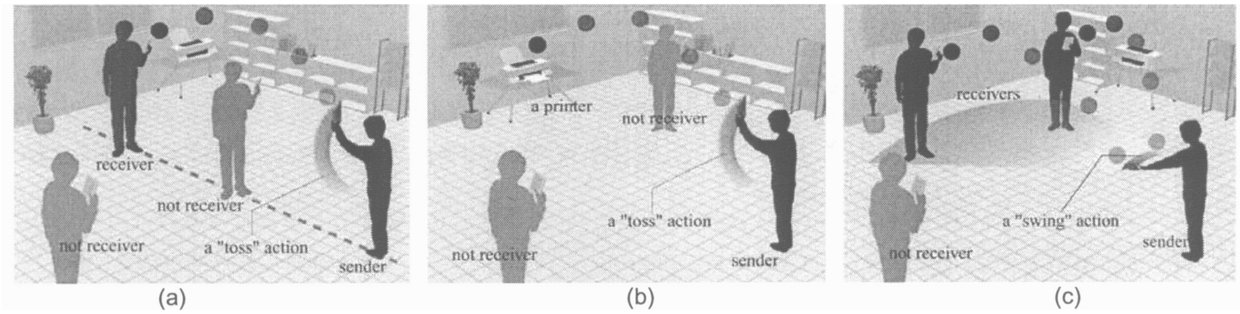

Figure 1. Intuitive information transfer techniques with *Toss-It*. (a) from a PDA to another PDA (b) from a PDA to a printer (c) from a PDA to multiple PDAs

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. 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 Figure 1(a) (unicast transfer).
- Print out an image from a user's PDA through a printer with just a "toss" action toward the printer (Figure 1(b)), or 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 multiple users with just a "(horizontal) swing" action toward them as shown in Figure 1(c). (multicast transfer).

We have developed a circuit board with inertial sensors and software executed on a server computer for recognizing a user's "toss" or "swing" action. Preliminary user studies of *Toss-It* were conducted to clarify how it could correctly identify users' actions.

2. Related Works

Several research projects have proposed techniques for information transfer. In [9], a special stylus pen with memories enables a user to transfer information from one device to another device in a "pick-and-drop" manner. [11] allows a user to conduct information transfer by pointing to the target device with RF pens and tags. 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 Figure 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 a move of a mobile device **to** provide more intuitive interfaces. For example, intuitive manipulations for mobile devices, such as maintaining viewing orientation when a device is rotated [5], or scrolling a screen by tilting a device [2], are proposed. XWand [12] 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 users' manipulations with it. [13] shows two-handed interaction techniques for a PDA by combining pen input and a spatially aware display. Several techniques for linking or manipulating multiple (mobile) devices have been proposed. SyncTap [10] links two devices by synchronously pushing a button or tapping a screen of each device. In [6], a user can create one large display from two tablet PCs or transfer information between them, by bumping or docking one of the PCs to the other.

One difference between these predecessor systems and *Toss-It* is that these systems are used for recognizing relatively slow or small moves of mobile devices (tilt or rotation), 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, while existing systems allow users to link between two devices. 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.

3. How to Transfer Information with "Toss" or "Swing" Actions

Figure 2 shows the system architecture of *Toss-It*. In order to realize intuitive information transfer techniques with "toss" or "swing" actions, *Toss-It* must satisfy the following requirements:

- *Toss-It* can recognize user's "toss" and "swing" actions conducted with his PDA.

- *Toss-It* can automatically identify the locations and orientations of multiple PDAs.
- 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

As for the third requirement, we use a wireless LAN and a server computer. The software on the server computer manages data on locations and orientations of multiple PDAs. The data is stored in a database management system as shown in Figure 2, and updated when changes on the locations and orientations of the PDAs have been detected. When a user "tosses" information with his PDA, the software module calculates the trajectory of the PDA, and identifies which devices the user wants to "toss" to, and sends the information to the devices via wireless LAN. As for the second requirement, we are now developing a mechanism for identifying locations and orientations of mobile devices (we discuss about it later in this paper). In this paper, issues related to the first requirement are described.

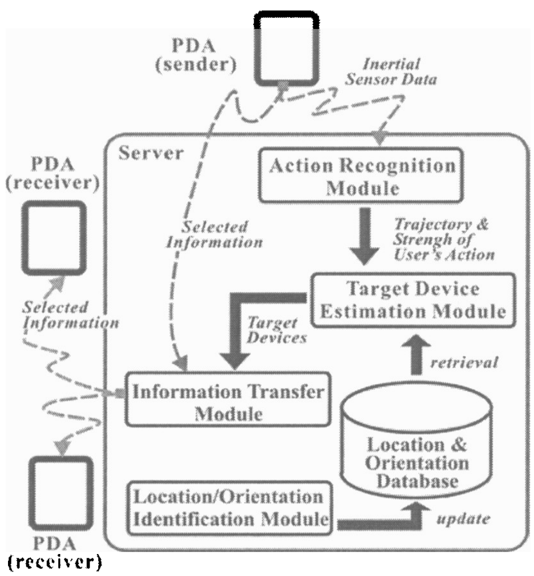

Figure 2. System architecture of *Toss-It*

4. Hardware Design

In order to recognize a "toss" or "swing" action, we have developed a circuit board that mounts accelerometers (Analog Devices ADXL210, ADXL202), gyroscopes (Murata ENC-03J, ENC-03M) and a microprocessor (Hitachi H8 microcomputer). The circuit board is designed to be attached to a PDA. Compared to other approaches, such as using multiple cameras for capturing real-time actions by multiple users in different locations and orientations, our approach requires lower cost (about $150 per circuit board), and no special equipment or setting, such as fixing cameras to a ceil or a wall of a room, and calibrating their installation angles. Therefore, users attach the circuit boards to their own PDAs, execute the software on a server computer, and then *Toss-It* is ready to start information transfer between the users' PDAs. In the circuit board shown in Figure 3, two 10g 2-axis accelerometers, two 2g 2-axis accelerometers and three I-axis gyroscopes are embedded. The 2g accelerometers are for capturing slow /weak "toss" or "swing" actions, while the 10g accelerometers are for capturing fast/strong actions. They are connected to a PDA through a serial communication via a microprocessor.

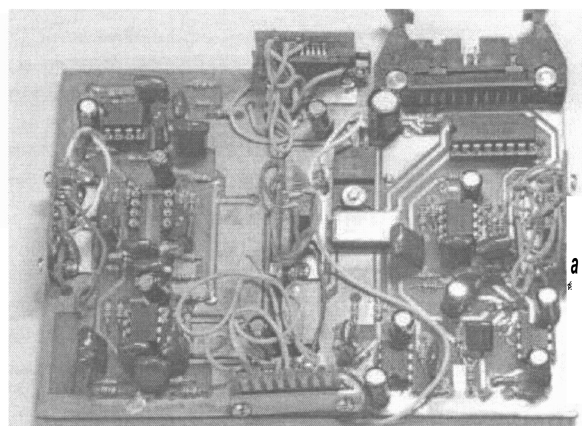

Figure 3. A circuit board with inertial sensors attached to a PDA

5. Recognition Algorithm

In order to identify 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 Figure 1(a)), and the trajectory of the "swing" action in a multicast situation (as shown in Figure 1(c))

5.1. Eliciting "Toss" or "Swing" Actions

Ideally, *Toss-It* can recognize a "toss" or "swing" action through the output data of the inertial sensors. We implemented a low-pass filter on our circuit board, in order to eliminate fluctuations of the output data. Several informal experiments, however, have indicated that non-negligible fluctuations still appear in the output data just after the action has been completed. Figure 4 shows a typical example of the output data of an accelerometer when a user has conducted a "toss" action with *Toss-It*.

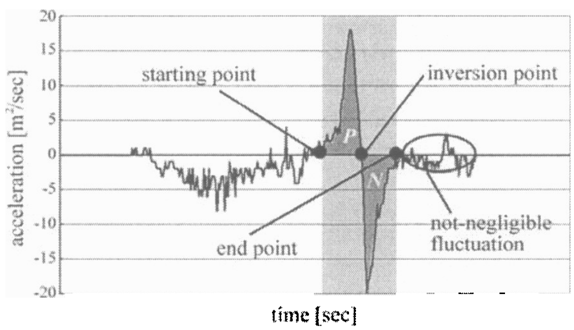

Figure **4. A** typical example of the output data of an accelerometer when a user conducted a "toss" action with *Toss-It*

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 Figure 4) is equal to that of the negative part *(N* in Figure 4).

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 Figure 4. 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 intersection 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 Figure 4).

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.

5.2. 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. Let v_0 be the maximum velocity, θ be the launch angle, and g be the acceleration of the gravity. *Toss-It* estimates the distance of the "toss" from the release point by the following expression:

$$distance = \frac{v_0^2 \sin 2\,8}{S}$$

5.3. Estimating the Trajectory of a "Swing" Action

Through a user's "swing" action, *Toss-It* identify devices as receivers of information as shown in Figure 1(c). In order to select receiver devices, it is necessary to calculate an angle that depicts how many degrees a user has swung his PDA around him. However, to calculate the angle is difficult, because a fulcrum point of a "swing action" is not determined precisely, because a location of a user is not identified precisely (although a location of the user's device can be identified precisely) and different types of "swing actions are possible (e.g. a swing by a flick of the wrist). Therefore, following assumptions are made to simplify the calculation: (a) a trajectory of a "swing" action is an arc, (b) we use only horizontal moves and neglect virtual moves of a "swing" action, and (c) the average length of a human arm as the radius of the arc is given (50 cm in the current version of *Toss-It).* An angle subtended by an arc is calculated by the following expression:

$$angle = 2\sin^{-1}\frac{l}{\boldsymbol{r}}$$

where l and $\boldsymbol{r}$ is the length and the radius of an arc, respectively.

6. Preliminary Experiments and Evaluations

We evaluated *Toss-It* on how accurately it could recognize receivers with "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.

Table 1. Average and standard deviation for each taraet distance

Target distance [m]	Average [m]	Standard Deviation
1	0.995	0.683
2	1.91	1.01
3	3.02	1.35

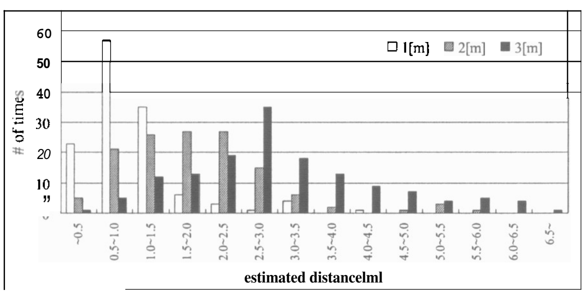

Figure 5. A distribution of the estimated distances

6.1. Results of the Recognition of a "Toss" Action

Figure 5 shows the distribution of the estimated distance of a "toss" by subjects. From this figure, we can find that there are peaks between 0.5 to 1.0 meters for subjects' one-meter-toss trials, those between 1.5 to 2.5 meters for their two-meter-toss trials, and those between 2.5 to **3.0** meters for their three-meter-toss trials. The average and standard deviation for each target distance are described in Table 1. Differences of "toss" actions between two target distances (1[m] and 2[m], or 2[m] and 3[m]) proved to be statistically significant by a t-test (two-tailed, p<.01).

6.2. Results of the Recognition of a "Swing" Action

Figure 6 shows the distribution of an estimated angle of a "swing" by the subjects. From Figure 6, we can find that there are peaks between 45 to 50 degrees for subjects' 45-degree-swing trials, those between 80 to 85 degrees for their 90-degree-swing trials, and those between **130** to 135 degrees for their 135-degree-swing trials. The average and standard deviation for each target angle are described in Table 2. Differences 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 t-test (two-tailed, p<.01).

The results of these experiments show that information transfer to electronic devices by "toss" and "swing" actions is possible, although the actions are not always recognized precisely. Comments from the subjects were positive, and favored this intuitive information transfer techniques.

Table 2. Average and standard deviation for each taraet angle

Target angle [deg]	Average [deg]	Standard Deviation
45	54.2	14.8
90	89.6	14.3
135	129	8.80

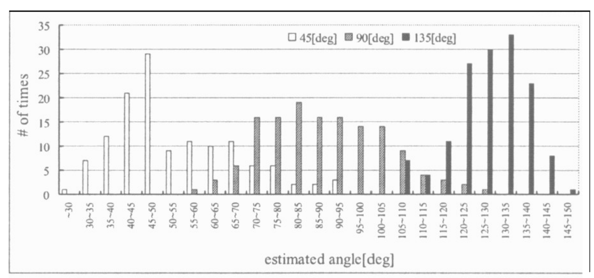

Figure 6. A distribution of the estimated angles

7. Issues Remain to be Investigated

There are several issues investigated in our future studies, in order to make use of *Toss-It* in a realistic setting.

7.1. Identifying Locations and Orientations of Devices

Identifying locations and orientations of devices automatically is the most critical issue in *Toss-It*. As studies on location-aware systems have attracted much attention recently [4], various approaches and systems for identifying locations and orientations of objects have been proposed.

In [7], locations and orientations of mobile devices are estimated based on measured RF signal strengths emitted from multiple base stations. As the IEEE 802.11b wireless Ethernet technology is popular and available for most mobile devices today, no extra equipment is necessary for implementing the proposed method. An estimation error of a device location is about one meter or so. In Active Bat system [1], multiple ultrasonic receivers are mounted at known points on a ceiling. A location of an object is calculated by the distances between a Bat (transmitter) attached to the object and each of the receivers. Multiple Bats are used for estimating an orientation of an object. The location estimation by Active Bat system is accurate to around a few centimeters. Cricket Compass [8] estimates locations and orientations of devices by using both RF and ultrasonic signals. It can determine orientations of devices to within 3 to 5 degrees when the true angle lies between ± 40 degrees. However, Cricket Compass cannot guarantee stable performance of its orientation identification, when the true angle is close to ± 90 degrees (its estimation error is more than 10 degrees). A computer vision approach is used for identifying locations and orientations of objects, for example, in the field of augmented reality research [3]. In many systems, a visual tag (marker) is attached to an object to be identified, in order to make the identification easier. However, a visual tag is often obtrusive, because, the tag itself is just an implementation matter, and not related to users' interests.

For successful information transfer by "swing" and "toss" actions, we set the following requirements for a location and orientation identification mechanism of *Toss-It*:

- An estimation error of a device's location is less than 10 cm.
- An estimation error of a device's orientation is less than 10 degrees.
- A sensor or a tag attached to a device is as small and light as possible so that a user can easily conduct a "toss" or "swing" action.

The first priority of our project in this phase is to confirm whether the idea of information transfer by a "toss" or "swing" action is practical and acceptable in the real world. To shorten the development duration of the identification mechanism, we have decided to use a computer vision approach (with a software library available through the Internet), although bothersome tasks such as installing and calibrating multiple cameras are required. The development is in progress and will be evaluated by the end of this year. When the idea of *Toss-It* has been proved to be practical and useful, more suitable mechanisms for *Toss-It* will be examined.

7.2. Information Transfer to Right Devices

Toss-It transfers information to a device whose location is closest to a calculated destination point by a user's "toss" action, or to devices whose locations are within a calculated angle by a user's "swing" action. However, information transfer with *Toss-It* is not always successful. Following are cases where information transfer does not work:

- Due to estimation errors of devices' locations and orientations, or recognition errors of strengths and trajectories of "toss" or "swing" actions, information is transferred to wrong devices.
- Due to errors of users' actions, information is transferred to wrong devices (we cannot always toss a physical object to an intended person or place).

Moreover, we have to consider the following cases, although users' actions is correct and *Toss-It* has correctly recognized their actions and devices' locations and orientations:

- 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 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 is required. However, from an HCI 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 decided 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 helshe 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 helshe wants to send the information to the device, otherwise touches the "reject" button.

Through the confirmation process, *Toss-lt* can avoid cases where a user sends information to unintended people or receives unwanted information. 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.

To reduce users' cognitive load for the confirmation process, it may be useful to use background knowledge. For example, when a user conducts a "toss" or "swing" action, receiver devices must be faced to the user, because *Toss-Zt* is used in co-located situations. Through user studies, we will evaluate what mechanism for the confirmation process is appropriate for *Toss-lt.*

7.3. *Toss-It* without a Server Computer

In the current version of *Toss-It,* a server computer is used to calculate a trajectory of a user's "toss" or "swing" action due to the limited computational capability of a **PDA.** However, if *Toss-lt* works without a server computer, it is desirable because of its less setup and maintenance tasks. When the performance of a mobile device has drastically been improved, it will be possible to execute the calculation on the device. At that time, by using a peer-to-peer communication technology for an ad-hoc wireless network, information transfer without a server computer will become possible.

8. User Scenarios with *Toss-Zt*

There are various user scenarios that *Toss-lt* makes possible. Figure **7** shows that user1 "tosses" a photo file in his **PDA** to user3 beyond user2. **All** user1 has to do is to select the photo file and conduct a "toss" action. In *Toss-lt,* users do not have to conduct bothersome manipulations on mobile devices, nor do users have to move close to a receiver of information.

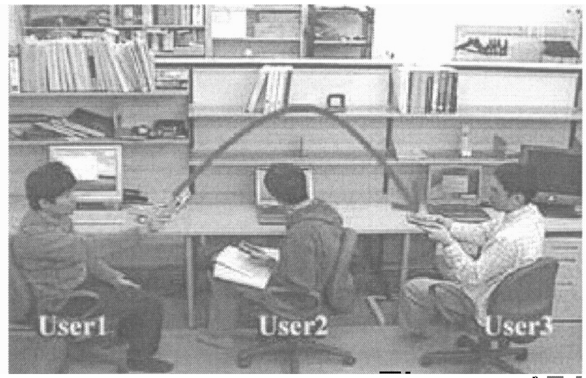

Figure 7. User1 "tosses" a photo to user3 beyond user2

9. Conclusions

In this paper, we described our research project in progress called *Toss-lt,* and discussed about its recognition mechanism for "toss" and "swing" actions. We confirmed the validity of our approach through the preliminary experiments and evaluations, and presented a user scenario with *Toss-Zt.*

The current version of *Toss-lt* assumes that locations and orientations of devices are given and fixed. We are developing a mechanism for automatically identifying their locations and orientations. We will then 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-lt* architecture, such as a universal remote controllers (for manipulating multiple devices simultaneously by a "swing" action, or a device behind an obstacle with a "toss" action), entertainment games, and so on.

References

[1] Addlesee, M. et al. "Implementing a Sentient Computing System", *IEEE Computer,* Vol. 34, No. 8, pp. 50-56, 2001

[2] Bartlett, J. "Rock 'n' Scroll Is Here to Stay", *IEEE Computer Graphics and Applications,* Vol. *20,* No.3, pp.45-50, 2000.

[3] Billinghurst, M., Kato, H., Poupyrev, I. "The MagicBook: Moving Seamlessly between Reality and Virtuality", *IEEE Computer Graphics and Applications,* Vol. 21, No. 3, pp. 2-4,2001.

[4] Hazas, M., Scott, J., and Krumm, J. "Location-Aware Computing Comes of Age", *IEEE Computer,* Vol. 37, No.2, pp. 95-97,2004,

[5] Hinckley, K., Pierce, J., Sinclair M. and Horvitz, E. "Sensing Techniques for Mobile Interaction", *In Proceedings of ACM UIST2000*, pp. 91-100,2000.

[6] Hinckley, K. "Synchronous Gestures for Multiple Persons and Computers", *In Proceedings of ACM UIST 2003,* pp. 149-158,2000.

[7] Ladd, A. et al. "Robotics-Based Location Sensing using Wireless Ethernet", *In Proceedings of MobiCom2002*, pp. 227-238,2002.

[8] Priyantha, P., Miu, A., and Teller, **S.** "The Cricket Compass for Context-Aware Mobile Applications", *In Proceedings of MobiCom2001,* pp. 1-14, 2001.

[9] Rekimoto, J. "Pick-and-Drop: **A** Direct Manipulation Technique for Multiple Computer Environments", *In Proceedings of ACM UIST'97*, pp. 3 1-39, 1997.

[10] Rekimoto, J. "SyncTap: **An** Interaction Technique for Mobile Networking", *In Proceedings of Mobile HCI,* 2003.

[11] Swindells, C., Tnkpen, K. M., Dill, J., C. and Tory, M. "That one there! Pointing to Establish Device Identity", *In Proceedings of ACM UIST2002*, pp. 151-160,2002.

[12] Wilson, A., and Shafer, **S.,** "Xwand: UI for Intelligent Spaces", *In Proceedings of ACM CHI2003*, pp. 545-552, 2003.

[13] Yee, K. Peephole Displays: Pen Interaction on Spatially Aware Handheld Computers, *In Proceedings of ACM CHI2003*, pp. 1-8, 2003.