Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

Original Paper

Point-Tap, Tap-Tap, and the Effect of Familiarity: to Enhance the Usability of See-and-Select in Smart Space

Seokhwan Kim,*1 Shin Takahashi*2 and Jiro Tanaka*2

Abstract – **We prototyped two selection techniques, Point-Tap and Tap-Tap, and conducted experiments to assess their characteristics, in particular how familiarity with a space affects their usability. Both techniques were developed to enhance the capability of the general "pointing gesture" and "map with live video" techniques. The goal of both techniques is to acquire a target object in smart space, and they share the concept of "see-and-select," which allows users to select an object while seeing the objects with their own eyes. Consequently, users must rely on the spatial locations of objects when using the techniques. According to spatial cognition science, humans recognize object locations in two ways, egocentrically and allocentrically, and some work has pointed out that users rely on allocentric representations more once they have become familiar with a space. Indeed, in our experiments, users who were familiar with the space could use the "map with live video" technique more effectively. The two main contributions of this paper are the presentation of the new techniques themselves, and the identification of a major factor for applying the techniques, namely, the users' expected familiarity with a space.**

Keywords : Pointing Gesture, Live Video, Map, Selection, Smart Space, Spatial Cognition, Egocentric, Allocentric

1. Introduction

Appropriate techniques that enable selection in smart spaces are necessary. In the near future, computers will be embedded everywhere and users will be able to interact with them. One basic and frequently required interaction in this environment is selection.

To interact with a device, users need to select it in advance. There are several techniques that enable selection, and we can classify them with respect to referable properties. Generally, when selecting an object, users need to refer to at least one property of the object, according to the features of the technique. For example, that property would be location when using a pointing gesture. With a command line interface, the property would be an identifier of the object. Table 1 classifies selection techniques according to such referable properties.

The concept of *see-and-select* is defined as *selection that can occur while users are seeing the objects with their own eyes*. This relies on spatial location (see Table 1) and can be beneficial, especially when there many similar objects. For example, there are typically several printers in offices and laboratories. When a new user needs to connect to a printer, which is usually near a given user, s/he must first find the printer. Currently, the user can find it through common interfaces that simply provide a long list. In this scenario, if the user can select the printer by its location (e.g., a printer in front of the user's own eyes), it will be more convenient. We expect that the need for such selection for devices and the benefits of see-and-select will increase as portable devices become more advanced, such as in selecting common displays or printers as peripheral devices for a smart phone.

Table 1 Selection techniques that rely on spatial locations and unique identifiers.

Techniques Relying on Spatial Location	Techniques Relying on Unique Identifier
Pointing gesture	Command line interface
Map with live video	Graphical user interface
	Speech recognition

"Pointing gesture" and "map with live video" are representative techniques of the see-and-select concept, and they have limitations in specific situations

*1: Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba
*2: Division of Information Engineering, Faculty of Engineering, Information and Systems, University of Tsukuba

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

[17] [20]. A pointing gesture is generally considered a natural and intuitive technique, but it can face occlusion problems if there are physical obstacles between users and objects. "Map with live video" shows the video from a camera that covers the whole range of the space and allows users to select an object shown in the video. A problem with this interface is that it can be hard to point out an object precisely if there are many objects and the screen size is small (i.e., a high density of objects).

To address these issues, we have designed and prototyped two techniques, called Point-Tap and Tap-Tap. Both techniques address the problems using a steerable camera in space. Users can change the direction of the camera with a pointing gesture or by tapping one point on the map with live video. Then, the image from the steerable camera is sent to the user 's mobile device and the user can complete the selection process.

We also conducted experiments regarding the characteristics of the new techniques, especially how familiarity with the space affects their usability. The techniques share the concept of see-and-select, and users need to rely on spatial locations. Thus, it is meaningful to find variables that affect spatial cognition in humans, and examinations considering the variables and techniques will help achieve the appropriate use of the techniques.

Humans recognize the locations of objects in space in two different ways: with egocentric and allocentric representations [11]. Egocentric representation allows the user to recognize locations with spatial relationships between him/herself and the objects. In contrast, with allocentric representation, users refer to spatial relationships between objects, as in maps. Klatzky pointed out that available representation differs according to familiarity with the space [10], and that people can rely on allocentric representations more once they become familiar with a space. Thus, selection techniques based on a pointing gesture and maps may have different effects with respect to familiarity with the space, and our experiments indeed showed a significant effect.

The remainder of this paper is organized as follows. Section 2 explains related techniques that enable see-and-select and describes the background of spatial cognition science. Sections 3 and 4 explain why and how the two techniques were prototyped, and discuss obtainable benefits. Section 5 describes user test procedures and presents our results. In Section 6, we provide our conclusions.

2. Related Work

In this section, we describe related selection techniques that use the concept of see-and-select, and also clarify why a pointing gesture and a map with live video were selected for this study..

2.1 Techniques for See-and-Select

2.1.1 Pointing Gesture

Users can select an object naturally with a pointing gesture if there is no barrier between the user and the object. A pointing gesture enables users to designate an object while looking at it with their own eyes. Hence, it is generally considered one of the most intuitive selection methods, and it is used in various domains, such as robotics [9], virtual reality [5], and smart spaces [20]. A pointing gesture system can be implemented using special types of wands [19] [20], laser pointers [12], or pure image processing with multiple cameras [21].

2.1.2 Mobile Augmented Reality (AR)

A mobile AR system can be considered one type of pointing-based selection, from an interaction-centric perspective, because it also relies on the direction of the device. A mobile AR system can track its direction and position with various technologies [4] [13] [14]. When showing the image from a camera placed in front of a mobile device, the system can calculate the point where an object is drawn on the screen with the tracked location and direction data. Consequently, the user can select an object while seeing it on the screen.

2.1.3 Map with Live Video

Map with live video has the benefit of avoiding the occlusion problem that can occur with pointing-based techniques, such as pointing gesture and mobile AR. *Sketch and Run* and *CRISTAL* are examples using this technique [16] [17]. They involve a camera installed in the middle of the ceiling, and show a video from the camera on digital surfaces. In both, a wide-view-angle lens is mounted on the camera and it shows the whole portion of the space with one image, as a general map does. Because the cameras are intentionally installed at optimized locations for the areas of interest, there is a low probability of facing physical barriers.

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

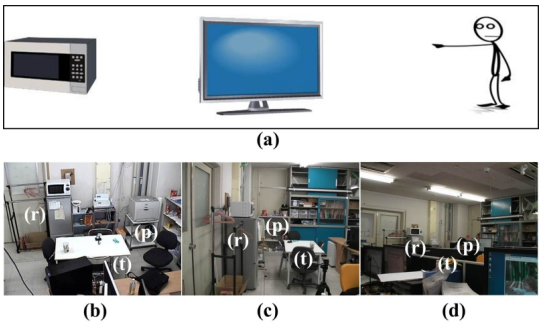

Fig. 1 Occlusion problems at different locations in a space.

2.1.4 Proximity-based Techniques

The proximity-based approach supports short-distance selection. In this approach, the selection occurs when the distance between a device and a controller is shorter than a certain threshold. Thus, it is not suitable for distant objects, which are not reachable by the user (e.g., selecting lights on a ceiling). This technique can be implemented in a cost-effective manner, such as by using visual codes or radio frequency identification (RFID) tags, and the implementation can be robust [1].

The two techniques addressed in this study are the pointing gesture and map with live video systems. Indeed, mobile AR and pure pointing gestures can be considered the same category (i.e., pointing-based), and both map with live video and pointing gesture can support the selection of distant objects. In contrast, users have to be near the object when using proximity-based techniques. This is why pointing gesture and map with live video were chosen for this study.

2.2 Human Recognition of Object Locations

To enhance the usability of the selection techniques described in the previous section, it is necessary to understand how humans recognize locations in space because users need to refer to locations. Humans recognize the locations of objects in two ways [11], as explained above. In *egocentric* representation, people memorize the location of an object by remembering the distance and direction from themselves to the object. Clearly, this representation is used when using a pointing gesture-based interface. In *allocentric* representation, people remember locations by recognizing spatial relationships between objects. For example, when memorizing the location of a display, people can remember it in terms of its relative distance and direction from other objects (e.g., the display on the left of the video player). Thus, when users have a map-like interface, they rely more on allocentric representations.

More allocentric representations are available to people who are familiar with the surroundings [10], which indicates that users who are familiar with the space can use a map-like interface more effectively than users who are not familiar with the space, because the map interface only provides allocentric representations. Thus, we may expect that the performances of the pointing gesture and map with live video systems may differ with respect to users' familiarity with the space. To investigate this, we conducted experiments with two user groups with different familiarities with the space. The results confirmed that familiarity had a significant effect.

3. Selection with Point-Tap and Tap-Tap

Two techniques were developed to enhance the capabilities of the pointing gesture and map with live video systems: Point-Tap and Tap-Tap. Here, we first describe the problems that motivated the design of the techniques.

3.1 Problems

Both systems can be problematic in specific situations. For example, when using the pointing gesture-based technique, it is hard for the user to make a selection when the object is occluded. Indeed, a pointing gesture is based on the assumption that the selectable objects are located within the user 's view. Thus, if there are physical barriers, or if selectable objects overlap each other, it is difficult for the user to select them.

Figure 1 illustrates the occlusion problems at different locations in a space. Figure 1a illustrates that a television hides a microwave from the user 's view. In Figure 1b - 1d, this same problem is shown with real images, where (r), (t), and (p) indicate a refrigerator, table, and printer, respectively. In Figure 1b, all objects are shown clearly. However, in Figure 1c and 1d some objects overlap each other or are hidden by other objects. Thus, it is difficult for the user to use a pointing gesture effectively.

A problem with the map with live video system is the density of selectable objects. When the density is high, it is difficult to point out an object pre-

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

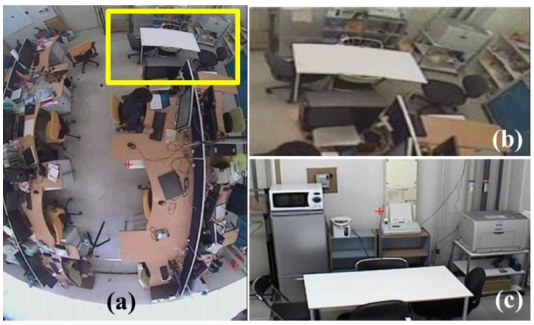

Fig. 2 Problem of naïve magnification of image from the camera with wide view angle lens.

cisely. This problem is worse when using a mobile device with a small screen. To address this, magnification can be used. However, with naïve magnification it is difficult to provide a natural view because a wide-view-angle lens typically distorts the image somewhat, especially at the edges. Another problem is that the viewing range is also limited by the capabilities of the lens.

Figure 2 illustrates problem of naïve magnification of an image from a camera with a wide-view-angle lens. Figure 2b shows magnified view of the yellow rectangle in Figure 2a. Figure 2c shows an image with a camera in the ceiling of the space, showing all objects in the area clearly, with an undistorted view. The image in Figure 2c is more natural than the one with naïve magnification shown in Figure 2b.

3.2 Exploiting a Steerable Camera on the Ceiling

The two problems described can be addressed using a steerable camera on the ceiling. Because the camera is installed on the ceiling and its direction is controllable, it can provide a broader view range and has a lower probability of facing occlusion problems. In this section, we explain how our two proposed techniques address these problems.

3.2.1 Point-Tap Interaction

Figure 3 illustrates the Point-Tap interaction. As shown in Figure 3a, a user makes a pointing gesture to the location of a target object. In the example, there are tables, a printer, and a fax. Then, the user can see an image of that location (Fig. 3b). To help identify objects, the system draws transparent rectangles with names over the selectable objects. Finally, the user can complete the selection by tapping one of rectangles.

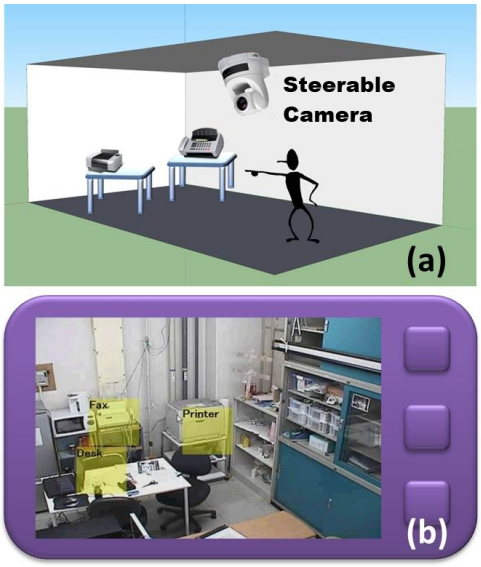

Fig. 3 Point-Tap interaction. After a user makes a pointing gesture to the location of the target object (a), s/he can see the image (b) on a mobile device.

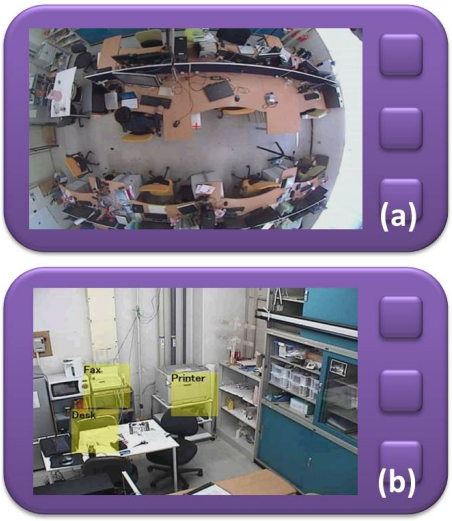

Fig. 4 Two different views for Tap-Tap; (a) For the first tap and (b) for the second tap.

Because Point-Tap uses a camera on the ceiling, the user can make a pointing gesture roughly, even if the object is not clearly visible (i.e., it can address the occlusion problem).

3.2.2 Tap-Tap Interaction

Tap-Tap initially shows an image from the camera with a wide-view-angle lens, which covers the whole range of the space (Fig. 4a). Then, instead of a pointing gesture, the user can tap a rough position on the image where the target object is seen. The user completes the selection in the same way as in

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

Table 2 Summary of related selection techniques based on see-and-select. Each method has advantages and disadvantages. With Point-Tap and Tap-Tap, it is possible to satisfy all attributes in the table.

	Pointing based [9] [13] [19] [20]	Proximity based [1]	Live Video on Tabletop [17]	Live Video on Mobile [6]	*Point-Tap*	*Tap-Tap*
Occlusion	hard to avoid	avoidable	avoidable	avoidable	avoidable	avoidable
High Density	hard to avoid	avoidable	avoidable	hard to avoid	avoidable	avoidable
Remote Selection	feasible	infeasible	feasible	feasible	feasible	feasible
Mobility	dependent	sufficient	limited	sufficient	dependent	sufficient

Dependent : It depends on the capability of tracking system.

Point-Tap. The mobile device shows the image (Fig. 4b), and the user can complete the selection by tapping one of the rectangles.

Table 2 compares Point-Tap and Tap-Tap to related techniques. Each technique has advantages and disadvantages. By adding one more live video view from the camera on the ceiling, the designed techniques can satisfy all of the attributes in Table 2.

3.3 Discussion of the Two Techniques

In this section, we discuss issues related to the techniques and explain the benefits that can be gained by using a camera on the ceiling.

3.3.1 Can Users Designate Hidden Objects Well?

The Point-Tap technique is designed to enable users to select objects even if the objects are hidden. Thus, it raises the question as to whether the user can designate the location of hidden objects well. Berkinblit et al. conducted an experiment to confirm this ability in humans [2]. Users were asked to make pointing gestures to hidden objects and made 5° angular errors at most in azimuth or elevation. This corresponds to about a 43 cm error when the distance between the user and the object is about 5 m. Thus, exact selection of a hidden object might be difficult, but the user can roughly designate its direction.

3.3.2 Live Video versus Rendered Map

Instead of an interface with live video, a rendered map image can be used. Indeed, a well-designed map image might be more understandable than a raw video in some scenarios. However, a live video can provide feedback immediately if the feedback is visually noticeable. Also, when considering authoring costs, a live video can be cheaper than a map. A map requires some authoring costs; however, a live video can be provided without additional cost if there is already a camera in the space. The cost for the map will also be higher when the layout of the space is changed frequently.

3.3.3 Occlusion-Free View

By exploiting a camera in the ceiling, the system has a lower probability of facing occlusion problems. Occlusion can occur when there are physical barriers between users and objects. In this situation, if the system can use other view sources, based on different locations, it has a higher probability of avoiding occlusion.

One issue is that the camera shows an image based on its location, which can be different from the view perceived by the users. Chan et al. and Rohs et al. conducted a series of experiments related to this issue [6] [15]. In the experiments in both papers, the users were able to overcome the different views.

3.3.4 Less Complexity in Object Recognition

Fixed and steerable cameras can recognize objects by referring to their geometrical locations. To enable selection through an image on a mobile device, the system should recognize the object. To achieve this generally requires unnatural markers [15]. One way around this is to try to recognize objects in a markerless way [3], but this is not easy to implement in terms of image processing and is prone to errors. In contrast, a fixed camera can detect objects by referring to geometrical positions rather than image processing. In addition, when an interactive object is moving, some other methods of object recognition are still necessary. However, there are many scenarios that deal with static fixed objects (e.g., large displays).

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

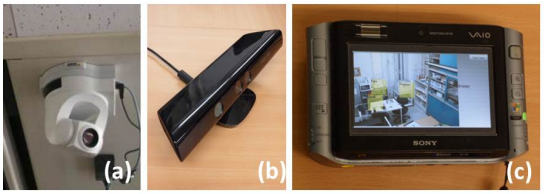

Fig. 5 Hardware for prototype system. Steerable camera (a), depth camera (b), and mobile device (c).

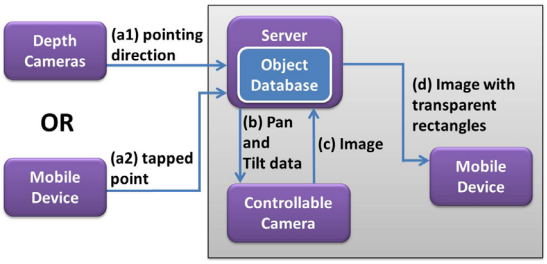

Fig. 6 Architecture and execution flow of Point-Tap and Tap-Tap.

3.3.5 Hand Jitter-Free

Another advantage of the view from a fixed camera is that it can be free from the hand jitter problem. When users rely primarily on the view from a mobile device 's own camera, hand jitter can cause the image to be unstable. Such systems need to provide methods for overcoming the jitter, using techniques such as freeze mode to stabilize images [4]. In contrast, with our setup, such stabilization is not required.

4. Implementation of Prototype System for Experiment

Here, we describe the implementation of a prototype system.

4.1 Hardware

For the steerable camera, an AXIS 214 [1] Network Camera was used (Fig. 5a). The direction of the camera can be adjusted by sending numerical values that represent pan and tilt through the network. The pan and tilt values describe how many degrees the camera is rotated horizontally and vertically, respectively. We used Microsoft 's Kinect [2] camera for tracking users ' pointing gestures (Fig. 5b). For the mobile device, we used a Sony UX50 [3] (Fig. 5c).

1: http://www.axis.com/products/cam_214
2: http://www.xbox.com/kinect
3: http://www.vaio.sony.co.jp/Products/UX1/feat1.html

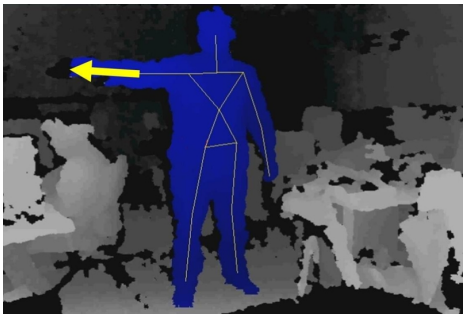

Fig. 7 Visualization of a tracked user. The blue area indicates the tracked user 's body. The system tracks the user 's pointing ray using the vector from the elbow to the hand (yellow arrow).

4.2 Overview

Figure 6 illustrates the overall architecture and execution flow of Point-Tap and Tap-Tap together. The difference between the two methods is the beginning of the execution. In Point-Tap, execution starts by sending a pointing direction (Fig. 6a1), but Tap-Tap starts by sending a tapped point (Fig. 6a2). Here, we provide an overview of the two techniques.

For Point-Tap, it is possible to track pointing direction by using depth cameras [18]. Tracked pointing data are sent to the server (Fig. 6a1), which starts to find the nearest object from the pointing direction. This is possible because the server maintains an object database that stores the location of each object. Then, the server sends the pan and tilt values of the detected object to the steerable camera (Fig. 6b), causing the camera to change its aim towards the designated object. The camera starts to take images and sends them back to the server (Fig. 6c). The server adds transparent rectangles for marking selectable objects and sends that image to the mobile device (Fig. 6d). Finally, the mobile device can display the image shown in Figure 3b.

The flow of Tap-Tap is essentially the same as Point-Tap, except at the beginning. When the user taps a point in the image (Fig. 4a), the tapped point is sent to the server (Fig. 6a2). The server finds the nearest object to the tapped point. The remainder of the process is the same as in Point-Tap (Fig. 6b-6d).

4.3 Tracking Pointing Gesture

It is difficult to track users 'pointing movements in different directions accurately if using only one camera [21]. To address this, two Microsoft Kinect depth cameras were used, and they were set up to face each

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)


```
<Space>
  <Object Name="Printer"
          PosX="532.3" PosY="-253.5" PosZ="1352.4"
          Pan="-25.5712" Tilt="19.65"/>
</Space>
```

Fig. 8 Example of object database in XML format.

other. The system can track a user 's movements when the user is placed between the two cameras.

To combine the two 3D spaces of the depth cameras, we arbitrarily picked three points in the shared area, and gathered the measured values, based on the two different coordinate systems of the cameras. Even if the measured values were different with respect to the location and orientation of the cameras, they were located in the same place. Thus, it was possible to calculate a matrix M that converts one camera space into another camera space by solving Equation 1 [8]:

$$M(x, y, z) = (x', y', z') \quad (1)$$

where (x, y, z) is the measured value in camera 1 and(x',y', z') is the measured value in camera 2

By using two depth cameras, the system can track a user 's pointing gesture in all directions. The system tracks the points of the hand and elbow, and the vector is used as a pointing ray. Figure 7 shows the visualization of a tracked user, and the yellow arrow illustrates a tracked pointing ray.

4.4 Server

4.4.1 Object Database

Figure 8 shows an example of an object database. The database is stored in XML format. The element " Object " contains six attributes. First, the name attribute is used to write the names of the objects with transparent rectangles (Fig. 3b). Attributes PosX, PosY, and PosZ are pre-measured positions of the object in the coordinate system of the camera. These position data are used to detect the nearest object from a tracked pointing ray. The attributes pan and tilt are used to specify the direction of the camera, as described above.

4.4.2 Marking Objects

As explained above, the system marks selectable objects with half-transparent rectangles. The system draws these by considering the pan and tilt values of all objects in the database. Because pan and tilt describe absolute directions from the camera, it is possible to determine whether the object is in view with respect to the current pan and tilt values (i.e., absolute direction) and the field of view of the camera. If the object is determined to be shown in the image, the system draws a rectangle with its name at the appropriate position.

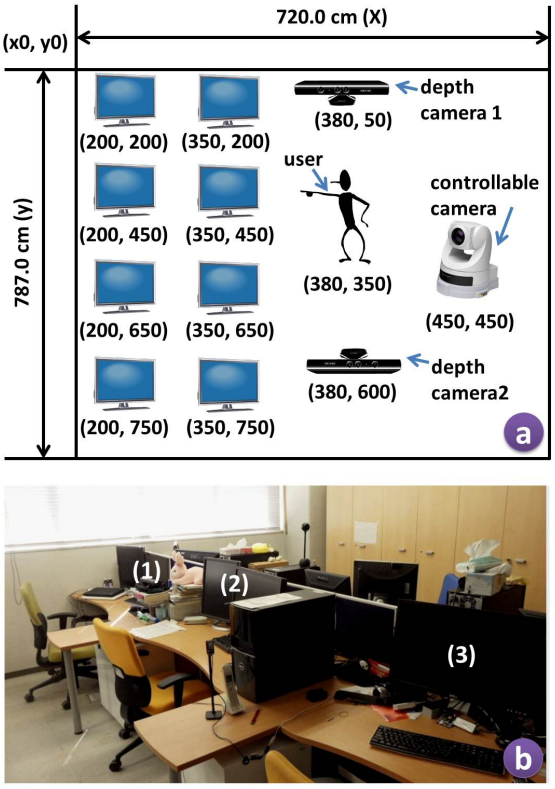

Fig. 9 Layout of hardware for the experiment. Target object installation (a) and real environment (b).

5. Evaluation

We conducted experiments using the prototype system described. There were two goals in the evaluation: *to confirm the effect of user familiarity with the space with the two techniques*, and *to compare the techniques to each other regardless of familiarity.*

5.1 Experiment Environment

5.1.1 Selection and Arrangement of Target Objects

Figure 9 shows the installation layout and real environment. In Figure 9a, the numerical values in parentheses below each object represent its location; the origin of coordinate is the top-left corner. The numerical values shown are exact but the locations of the objects in the figure were adjusted to include all objects in one image. We selected eight objects

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

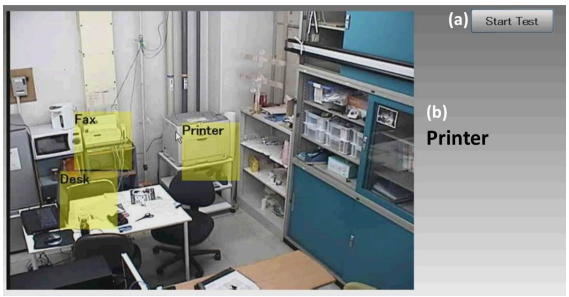

Fig. 10 The application used for the experiments with both techniques.

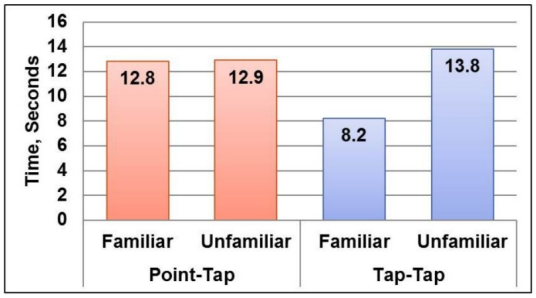

Fig. 11 Effect of familiarity with the space. In Point-Tap, familiarity did not have a large effect. In contrast, in Tap-Tap, the users in the unfamiliar group took significantly longer.

for the experiments, all of which were displays. This was intended to provide the same difficulty in distinguishing objects. For example, if users are asked to select an object such as a television or microwave, they can distinguish between them by external shape rather than by location. Thus, we selected similar objects (i.e., displays). This is also a scenario where see-and-select is more beneficial, where it is easier to distinguish objects by location than by other properties (e.g., identifier or external shape).

Figure 9b shows the real environment. Displays were located on top of desks (Fig. 9b1 - 9b3), and the desks were used by real users. For the experiments, we did not adjust the original layout of the space. This was because users who are familiar with a space can become unfamiliar if the objects are arbitrarily moved for a test. To distinguish between displays, we used each owner 's name (e.g., Nicole 's display or Scott 's display). When there were multiple displays on a desk, we asked users to designate the middle of the displays.

5.1.2 User Groups

We recruited two different user groups. The first group used the experimental space every day. Thus, they were familiar with the space (*familiar group*). The second group visited the space fewer than five times in the 1 month before the test date (*unfamiliar group*). There were seven users in each group. In total, 14 users between the ages of 24 and 30 years (average, 26 years) participated. There were 12 males and 2 females.

5.1.3 Tasks and Measured Values

For the experiment, the application shown in Figure 10 was developed. When the user pushes the start button (Fig. 10a), the system shows the name of a target object (Fig. 10b). After the selection, the name of the next target object is shown automatically. We asked the users to make selections for each object using Point-Tap and Tap-Tap. The order of the selection for objects was identical for all users and both types. Before the test, the users had 10 min to practice both methods. In this time, the users who were not familiar with the space were asked to memorize the locations of the objects. For all cases, we measured the time taken to complete the selection.

5.2 Result

5.2.1 Effect of Familiarity

Familiarity had a significant effect on Tap-Tap but not on Point-Tap. Figure 11 shows graphs illustrating the effect of familiarity on Point-Tap and on Tap-Tap broken down by group. The groups performed similarly for Point-Tap, where the familiar group took 12.8 s and the unfamiliar group took 12.9 s, on average. One-way analysis of variance (ANOVA) showed that $F(1,14) = 0.0006$, $P = 0.97$. In contrast, the results for Tap-Tap were quite different; the familiar group took 8.2 s, but the unfamiliar group took 13.8 s. An ANOVA test showed a significant effect, with $F(1,14) = 13.37$, $P = 0.002$.

5.2.2 Point-Tap versus Tap-Tap

The two techniques showed similar performances, but more users preferred Point-Tap than Tap-Tap. As shown in Table 3, Point-Tap and Tap-Tap took 11.54 and 10.52 s, respectively, on average. Point-Tap took slightly longer, probably due to the time taken for the physical movement of the pointing gesture. After the test, we asked users to select their preferred method. Ten users (71%) responded that they preferred Point-Tap over Tap-Tap, and most of

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

Table 3 Performance was not significantly different. However, 71% of users preferred Point-Tap over Tap-Tap.

Method Type	Time (s)	Preference
Point-Tap	11.54	10 users (71%)
Tap-Tap	10.52	4 users (29%)

them reported that it felt more intuitive and natural to make a pointing gesture.

5. 3 Discussion

Users ' familiarity with the space had a significant effect on Tap-Tap but not on Point-Tap. This result is consistent with findings in spatial cognition science. This suggests that, with Point-Tap (Fig. 11), even users who were not familiar with the space had no problem remembering the locations well. This is because they were remembering the locations of the objects with egocentric representations, i.e., the direction and distance from themselves to the objects, and thus they had no significant problem using a pointing gesture because it relies on the same representations. In contrast, with Tap-Tap, the users needed to remember the locations with allocentric representations, but this type of representation is not readily available until users have become familiar with the space [10]. Thus, familiarity with the space had a significant effect on Tap-Tap.

In general, males have better spatial cognition [7]. In our test, there were two female participants and they were in the unfamiliar group. We compared the results of the two females to the results of the male participants in the unfamiliar group, and found no significant difference. With Point-Tap, male and female users took 13.8 s and 12.1 s, respectively, on average. With Tap-Tap, they took 13.2 s and 15.2 s. Hence, female users showed a slightly faster speed with Point-Tap and a slower performance with Tap-Tap. We conducted a one-way ANOVA test with the averaged results of Tap-Tap, but there was no significant effect ($F(1,14) = 0.81$, $P = 0.38$).

Table 4 summarizes the choices with consideration of the expected user 's familiarity and higher priority of the space. When user satisfaction is more important, Point-Tap will be more promising for both types of user (familiar and unfamiliar with the space). This is because 71% of users preferred Point-Tap over Tap-Tap (see Table 3). When effectiveness has a higher priority, Tap-Tap will be better if the users are familiar with the space because users in the familiar group took less time with Tap-Tap than Point-Tap (Tap-Tap: 8.2 s, Point-Tap: 12.8 s; see Fig. 11). However, if the users are unfamiliar with the space, Point-Tap will be more promising because they took 12.9 s with Point-Tap but 13.8 s with Tap-Tap (see Fig. 11).

Table 4 Summary of beneficial places for Point-Tap and Tap-Tap. If the users are expected to be familiar to the space and the effectiveness is more important, Tap-Tap will be more promising. Otherwise, Point-Tap will be more appropriate.

	Familiar	*Unfamiliar*
Satisfaction	Point-Tap Ex) Staff lounge	Point-Tap Ex) Museum
Effectiveness	Tap-Tap Ex) Office	Point-Tap Ex) Lecture room

An important factor for a usability test is the failure rate. In these experiments, there was no case of failure. In the experiment, users were asked to make a rough pointing gesture (or tap in Tap-Tap) and complete the selection by picking one rectangle on the mobile device screen. These were relatively easy tasks and the participants were hard to make failure cases without mistakes. The density of target objects was also low. A maximum of four objects was shown at the same time and there were no overlaps among objects. This is probably why there was no case of failure. We expect that higher densities would affect the techniques differently, and further investigations in such an environment should be conducted.

6. Conclusions and Future Work

Point-Tap and Tap-Tap are based on the pointing gesture and a map with live video techniques, respectively. The new techniques were designed to enhance the capabilities of both the pointing gesture and a map with live video systems, and we verified the concept by comparing them to related techniques. In experiments a usability factor was identified, namely, the users 'expected familiarity with the space. The results show that familiarity significantly affected Tap-Tap, indicating that the expected user 's familiarity with the space should be considered when

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

choosing between pointing gestures and map-based interfaces as a selection method.

A camera on the ceiling takes time to move, and this can result in slow performance with both techniques. However, that physical time was assessed for all test participants for both methods in the same way; thus, the statistical data allow for a meaningful comparison of the techniques. In future work, we plan to use a higher resolution and a camera with a wider angle of view. Then we will be able to gather a natural view that shows different areas in space through fine image processing. We expect that the speed of the techniques will be improved with this setup.

In this study, we considered only the case that a user and the selectable objects are in the same space. However, the two techniques can be applied to a scenario with a distant remote space, because they rely on the view from the camera. Finding different usability factors with that setup may be a promising future direction.

Reference

[1] Ailisto, H., Plomp, J., Pohjanheimo, L., Strommer, E.: A Physical Selection Paradigm for Ubiquitous Computing, Ambient Intelligence, *Lecture Notes in Computer Science*, Vol. 2875, pp.372-383 (2003).

[2] Berkinblit, M. B., Fookson, O. I., Smetanin, B., Adamovich, S. V., Poizner, H.: The interaction of visual and proprioceptive inputs in pointing to actual and remembered targets, *Experimental Brain Research*, Vol. 107, No. 2, pp.326-330 (1995).

[3] Ballagas, R., Borchers, J., Rohs, M., Sheridan, G. J.: The Smart Phone: A Ubiquitous Input Device, *IEEE Pervasive Computing*, Vol. 5, No. 1, pp.70-77 (2006).

[4] Boring, S., Baur, D., Butz, A., Gustafson, S., Baudisch, P.: Touch projector: mobile interaction through video, Proceedings of the ACM international conference on Human factors in computing systems, pp. 2287–2296 (2010).

[5] Bowman, D. A., Johnson, D. B., Hodges, L. F.: Testbed evaluation of virtual environment interaction techniques, Proceedings of the ACM symposium on Virtual reality software and technology, pp.26-33 (1999).

[6] Chan, L., Hsu, Y., Hung, Y., Hsu, J. Y.: A Panorama-Based Interface for Interacting with the Physical Environment Using Orientation-Aware Handhelds, The Seventh International Conference on Ubiquitous Computing (2005).

[7] Greary, D. C., Saults, S. J., Liu, F., Hoard, M. K.: Sex Differences in Spatial Cognition, Computational Fluency, and Arthmetical Reasoning, Journal of Experimental Child Psychology 77, pp. 337-353, Academic Press (2000).

[8] Foley, J. D., Dam A. V., Feiner, S. K., Hughes, J. F.: Chapter5. Geometrical Transformations, Computer Graphics Principles and Practice, pp. 212-228, Addison Wesley (1997).

[9] Kemp, C. C., Anderson, C. D., Nguyen, H., Trevor, A. J., Xu, Z.: A point-and-click interface for the real world: laser designation of objects for mobile manipulation, Proceedings of the ACM/IEEE international conference on Human robot interaction, pp. 241-248 (2003).

[10] Klatzky, R. L.: Allocentric and Egocentric Spatial Representations: Definitions, Distinctions, and Interconnections, *Lecture Notes in Computer Science*, Vol. 1404, pp.1-18 (1998).

[11] Meilinger, T., Vosgerau, G.: Putting egocentric and allocentric into perspective, Proceedings of the 7th international conference on Spatial cognition, pp. 207–221 (2010).

[12] Patel, S. N., Abowd, G. D.: A 2-Way Laser-Assisted Selection Scheme for Handhelds in a Physical Environment, Ubicomp 2003, *Lecture Notes in Computer Science*, Vol. 2864, pp. 200-207 (2003).

[13] Patel, S. N. Rekimoto, J., Abowd, G. D.: iCam: Precise at-a-Distance Interaction in the Physical Environment, *Pervasive Computing*, pp. 272-287 (2006).

[14] Pears, N., Jackson, D. G., Olivier, P.: Smart Phone Interaction with Registered Displays, *IEEE Pervasive Computing*, Vol. 8, No. 2, pp. 14-21 (2009).

[15] Rohs, M., Schoning, J., Raubal, M., Essl, G., Kruger, A.: Map navigation with mobile devices: virtual versus physical movement with and without visual context, Proceedings of the ACM international conference on Multimodal interfaces, pp. 146-153(2007).

[16] Sakamoto, D., Honda, K., Inami, M., Igarashi, T.: Sketch and run: a stroke-based interface for home robots, Proceedings of the ACM international conference on Human factors in computing systems, pp. 197-200 (2009).

[17] Seifried, T., Haller, M., Scott, S. D., Perteneder, F., Rendl, C., Sakamoto, D., Inami, M.: CRISTAL: a collaborative home media and device controller based on a multi-touch display, Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, pp. 33-40 (2009).

[18] Villaroman, N., Rowe, D., Swan, B.: Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor, Proceedings of the ACM conference on Information technology education, pp.227-232 (2011).

[19] Wilson, A., Hubert, P.: Pointing in Intelligent Environments with the WorldCursor, Proceedings of IFIP Interact, pp. 495-502 (2003).

[20] Wilson, A., Shafer, S.: XWand: UI for intelligent spaces, Proceedings of the ACM international conference on Human factors in computing systems, pp. 545–552 (2003).

[21] Yamamoto, Y., Yoda, I., Sakaue, K.: Arm-Pointing Gesture Interface Using Surrounded Stereo Cameras System, Proceedings of the Pattern Recognition, pp. 965-970 (2004).

(received Feb. 2, 2012, revised Aug. 2)

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)

Biography

Seokhwan Kim

Seokhwan Kim is a PhD candidate in computer science at University of Tsukuba. His research interests include novel interaction technologies, ubiquitous computing, and sustainable HCI. He received a BS in software engineering and a MS in computer science at Sangmyung University in 2007 and 2009 respectively.

Shin Takahashi

Shin Takahashi is an Associate Professor of Department of Computer Science, University of Tsukuba. His research interests include user interface software and ubiquitous computing. He received his BSc, MSc, and PhD in Information Science from the University of Tokyo in 1991, 1993, and 2003. He is a member of ACM, IPSJ, and JSSST.

Jiro Tanaka

Jiro Tanaka is a Professor of Department of Computer Science, University of Tsukuba. His research interests include ubiquitous computing, interactive programming, and computer-human interaction. He received a BSc and a MSc from University of Tokyo in 1975 and 1977. He received a PhD in computer science from University of Utah in 1984. He is a member of ACM, IEEE and IPSJ.

Information and Media Technologies 8(1) : 97–108 (2013)
reprinted from: The Transactions of Human Interface Society 14(4) : 143–154 (2012)