iThrow : A NEW GESTURE-BASED WEARABLE INPUT DEVICE WITH TARGET SELECTION ALGORITHM

JONG-WOON YOO, YO-WON JEONG, YONG SONG, JUPYUNG LEE,
SEUNG-HO LIM, KI-WOONG PARK, AND KYU HO PARK

Computer Engineering Research Laboratory
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
{jwyoo, ywjeong, ysong, jplee, shlim, woongbak, kpark}@core.kaist.ac.kr

Abstract:

We present an intelligent interface system which includes a new gesture-based wearable input device, called *iThrow*, as a main user interface for mobile devices, and an infrastructure helping users be aware of and make a use of various public devices in user-friendly manners. In this kind of intelligent interface system, selecting an object among multiple ones is one of the fundamental functions because it is a pre-cursor to all other subsequent actions. We propose a new selection algorithm which improves selection speed by adaptively resizing the objects' angular widths. Results show that the proposed algorithm outperforms the ray-based selection technique in selection speed about 62.6%.

Keywords:

Intelligent interface system; gesture-based input device; selection technique; intuitive interface; user-friendly interface;

1. Introduction

In recent years, the rapid progress of ubiquitous and pervasive computing technology has led to the emergence of various 'smart' places ranging from a smart room to a smart city. The smart spaces mainly aim at providing a communication channel between users and computing resources including many devices. Our ambitious project, called *U-TOPIA*, has developed *Ubiquitous Fashionable Computer* (*UFC*) as a personal computing environment and has been actualizing a campus-wide smart space [1]. The ultimate goal of this project is that users with mobile computing devices including UFC are able to communicate with each other and utilize various ubiquitous service devices within the campus. As an HCI related research in U-TOPIA, we are currently involved in an intelligent interface system that allows users to interact with various devices or other users in user-friendly manners such as simple and recognizable hand gestures.

In this paper, we present an intelligent interface system which includes a new gesture-based wearable input device, called *iThrow*, as a main user interface for mobile devices and an infrastructure helping users be aware of and make a use of various public devices such as printers, monitors. *iThrow* is a ring-type wireless input device which is small enough to be worn on one's finger. A user wearing this device can select one of the public devices with an intuitive pointing action and manipulate it with simple hand gestures. Meanwhile, in order to understand his gesture and map the devices into a virtual space where each device is expressed as a specific sized rectangle, our location server keeps the information of the devices and its middleware has them operate according to the gesture command. In this paper, we particularly focus on the target selection for use and we propose a new selection algorithm that improves selection performance not only for our interface system but for other virtual environments.

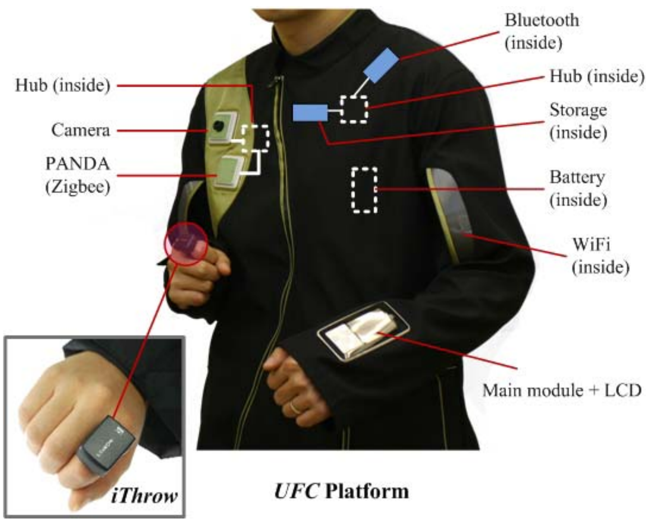


Figure 1. UFC and *iThrow*

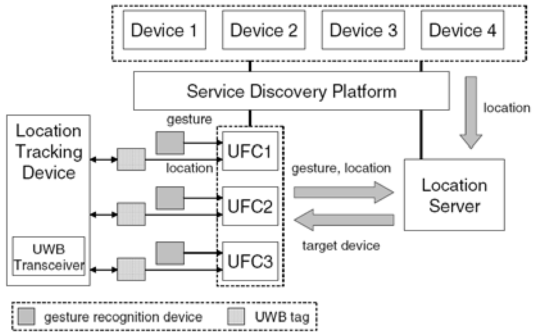


Figure 2. Overall system architecture

2. *iThrow* System

iThrow system provides a user-friendly way to interact with ubiquitous computing environment. Each person uses UFC as a main user computing device in our system.

2.1. Motivation

Due to its small form factor, most portable devices, including our UFC, have only small-sized display and limited input devices. UFC has 2.5” LCD display at the sleeve of the clothe and 12 input buttons that is shown in Figure 1, which are definitely insufficient to easily use UFC such as to monitor the status of main and various peripheral modules of UFC, control the modules, and send a user’s intention to UFC.

This problem is exacerbated when a UFC user tries to select and control various ubiquitous devices using one’s UFC: as the number of controllable ubiquitous devices increases, it becomes more difficult to target a desirable device among them and exchange information with it, due to the small-sized display and limited input devices of UFC. Efficient utilization of a small-sized display and intelligent mapping of various commands on the input buttons can only partially address this problem. However, such an approach usually makes it difficult to learn the usage of the device, which degrades the usability of UFC. One recent workshop underscored that usability is one of the primary challenges in a next-generation “smart” room where is full of various ubiquitous devices [2].

We attempt to resolve this problem by making full use of spatial resources inside the testbed room: given that various ubiquitous devices are spatially distributed inside the testbed room, a UFC user can easily represent his intentions through his spatial movement and gestures. For example, let us assume that one UFC user intends to put a picture he takes on a public display and other people can see the picture. From the perspective of the user, the most natural way of reflecting his intention on the related ubiquitous environment is pointing his finger at the public display and throwing the picture at the public display. By employing this kind of a user-friendly and gesture-based interface and fully utilizing abundant spatial resources, the limitation of UFC can be overcome.

2.2. System components

The overall system to support user-friendly and gesture-based interface is composed of the following components:

A. *iThrow* is a ring type wearable input device which recognizes a user’s gestures and pointing directions.

B. *Location tracking system* keeps track of the location of users’ and public devices in the smart space. This system is essential because the absolute location information of the users and the public devices is critical information to find the target devices that users point at. We utilized a UWB-based location tracking system [8] whose typical accuracy is 6 inches (15cm).

C. *Location server* gathers and manages the location information from the location tracking system. The main role of location server is to identify which device is pointed by each user using the location information. When a user points at a device, the pointing direction recognized by *iThrow* is transferred to the location server and the location server finally decides what the pointed device is.

D. *Service discovery platform*: For UFC to exchange information with any ubiquitous device, the UFC should be able to discover the available communication interface including IP address, port number and various properties of the device. For developing the ubiquitous service discovery (USD) protocol, we have been working with a middleware expert team, and they developed an efficient USD protocol as a part of μ-ware based on KUSP (KAIST Ubiquitous Service Platform) [3]. The USD protocol was originally based on UPnP [7], which is widespread as a service discovery, and this protocol is simplified to avoid XML parsing overhead. In this study, the USD protocol on μ-ware was used as the service discovery platform.

Overall architecture is illustrated in Figure 2. Among these components, in this paper, we focus on *iThrow* and the target selection algorithm running on the location server.

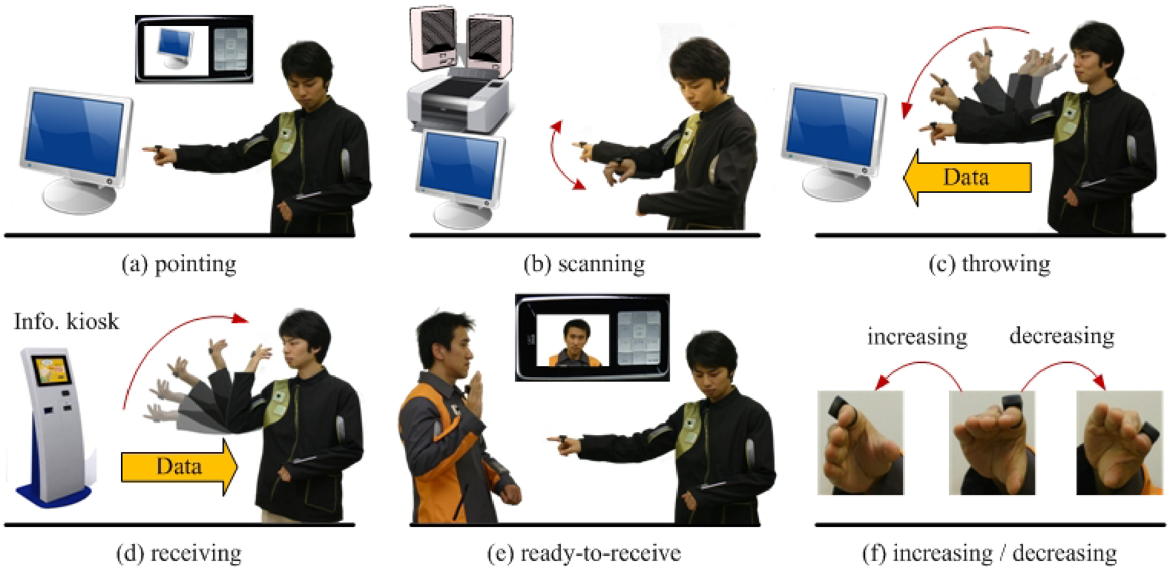


Figure 3. Gesture sets of *iThrow*

3. *iThrow*

iThrow is a ring-type wearable input device which is small enough to be worn on one's finger. It has a three-axes accelerometer [9] and a three-axes magneto-resistive sensor [10].

Using both sensors, we can get the orientation of *iThrow* [4]. Therefore, we can get the pointing direction by calculating the orientation of *iThrow* whenever the user points at a certain device.

The accelerometer is also used for recognizing the user's hand gestures. We defined several hand gestures and summarized them in Figure 3. Because we have the limitation of space and the gesture recognition is not the focus of this paper, we omit the detailed explanation of gesture recognition algorithm.

Every time a user points at a device, UFC displays the selected target device upon its screen. The scanning gesture allows a user to investigate controllable devices inside the room. This scanning operation is similar to the operation of moving a mouse pointer across several icons in a typical PC desktop environment. 'Throwing/Receiving' gesture is used to send/receive data to/from the others. 'Ready-to-receive' gesture is necessary for a UFC user to express one's intention to receive other UFC users' objects. When one user makes a pointing or scanning gesture, only limited users who take the 'ready-to-receive' gesture can be selected.

4. Target selection

Target selection is the main role of the location server that identifies which device is pointed by the user. In this chapter, we explain our target selection approach used in *iThrow* system.

4.1. Target searching with graphical feedback and its problems

In current version of *iThrow* system, the real space is projected onto a 2-dimensional virtual space. Within the real space, many of public devices are deployed and each of them can be described in the virtual space as a rectangular of which size is proportional to its real size. A user is represented in the virtual space as a circle and the size of the circle depends on his/her body size. The location server communicates with the location tracking system to keep the location information up-to-date.

We first took the ray-based minimum angle selection as a naive approach which is described in Figure 4. When a user points at a device with *iThrow*, a pointing direction is measured and sent to the location server to identify which device is pointed by the user. The location server casts a ray from the user toward the pointing direction in the virtual space. Then it selects a device which is the closest to the ray. The 'closest' here means that the included angle between the casted ray and a device is minimal. When two or more devices have the same minimal angle (B and C in Figure 4), the device whose distance from the user is minimal is selected. If all of the included angles are bigger than a pre-defined threshold angle A_{mar}, no selection occurs.

With his pointing action, he can search devices capable to interact. Figure 5 shows how to search devices. When a user wearing UFC points at a device with *iThrow*, it measures the pointing direction, the location server then finds the target device according to the direction. After that, he can see the selected device with its information on the screen of UFC. If the one wanted by him is not selected at the first time, he is noticed from the screen that currently the wrong one is selected and then, he will retry to point with a little hand movement. Therefore, he can finally select a device with the graphical feedback.

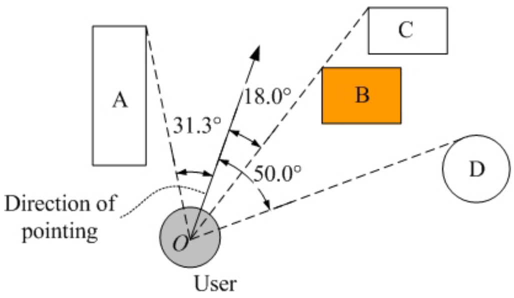


Figure 4. Ray-based minimum angle selection

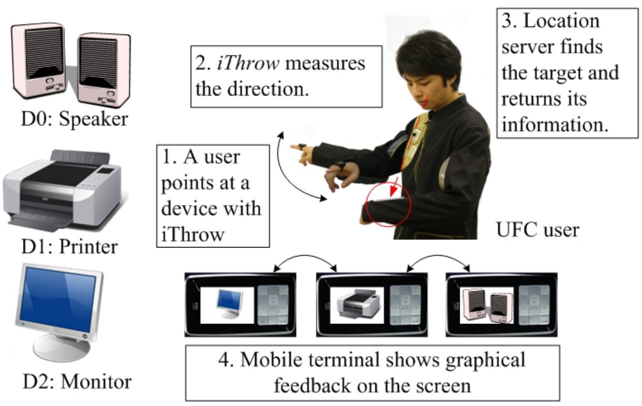


Figure 5. Graphical feedback helps the user to select the target device correctly.

The feedback mechanism is so expedient to provide the reliability to users and makes the wrong selection correct. Due to many sources of miss-pointing including errors of magnetic sensors and location tracking, hand trembling, jitters, and even user induced mistakes, wrong selection happens frequently. But, by virtue of the feedback, the user can correct the miss-pointing.

For example, in Figure 5, let's assume that he wants to select device 1, printer. Even if he thinks that he points it correctly, because of the miss-pointing errors, he can be announced through the graphic feedback that the left-side monitor is selected. Then, he tries to select the printer by moving his hand to the right. If the right-side speaker is, unfortunately, selected instead of the printer, he has to move his hand again. He can finally point at the printer by interacting with the feedback. However, if the printer places too close to the other devices, it is harder for him to select it. How about in case of his desired device is very small? These cases can make the system useless.

The difficulty of pointing or selecting a target device is closely related to its physically assigned angular width, which depends on the device size, its relational location, and the user's location. Fitts' Law describes well the relationship of the assigned angle and the difficulty with selection [5][6]. According to the law, the index of difficulty is expressed as a log function of angular movement and angular target width and it is proportional to the time for selection.

The Fitts' law is confirmed by our experimental result with *iThrow* which is introduced at the experiment section.

4.2. Adaptive angle assignment

As mentioned at the previous section, the small angular width of a device makes a user get in trouble with selecting it. As a result, it takes more time to select. Hence, we insist that a threshold of the angular width should be guaranteed. The threshold value makes it possible to select the device within a reasonable time.

For this, we propose an algorithm, called adaptive angle assignment, which makes all of the assigned angular width of devices from the user in a given space bigger than the specific threshold, A_{TH}. This algorithm solves the problem of the lagged selection time by reassigning the angular width. When he starts the target selection at the specific location, the location server calculates the physically assigned angular widths of the devices and then, if necessary, it reassigns adaptively the angular widths. The process is following.

A. *Grouping of target devices*: After the calculation of physically assigned angular width, the location server creates an angle table as shown in Figure 7. The contiguous angles are regarded as a group angle and the location server reassigns each angle within a fixed group angle.

B. *Adaptive angle assignment*: The server performs the reassignment for each group according to our algorithm which is represented as a pseudo code in Table 1.

G_k is k^{th} group angle and A_i is i^{th} angle in each G_k. A_{lack}

is a sum of required angles for expanding angles, which are smaller than A_{TH}, to A_{TH}. A group with zero of A_{lack} is not necessary to be reassigned. A_{res} is a sum of excessive angles to A_{TH} within a group. That this value is zero means that the group cannot reassign. In addition, A_{don} is a sum of the angles which are donated from the excessive angles. If A_{res} is smaller than A_{lack}, A_{don} becomes A_{lack} and otherwise, it becomes A_{res}. After deciding the value of A_{don}, all angles are expanded or shrunken with a proportion to the gap angle with A_{TH}. At this time, the group angle is consistent because the expanding angle and the shrunken angle are same. Figure 8 shows the reassignment result when A_{TH} is set to 10 degrees.

Table 1. Adaptive angle assignment

For all A_i in G_k {
    If $|A_i| < A_{TH}$ then A_{lack} += $(A_{TH} - |A_i|)$;
    Else then A_{res} += $|A_i| - A_{TH}$;
}

If $A_{lack} > A_{res}$ then $A_{don} = A_{res}$;
Else then $A_{don} = A_{lack}$;

For all A_i in G_k {
    If $A_i < A_{TH}$ then
        $|A_i|$ += $A_{don} \times (A_{TH} - |A_i|) / A_{lack}$;
    Else then
        $|A_i|$ −= $A_{don} \times (|A_i| - A_{TH}) / A_{res}$;
}

Even though a little gap between the original angular region, which is a region between start degree and end degree, and the reassigned angular region exists, it does not affect the performance of selection much because of general operating pattern of users; when a user wants to select a device with *iThrow*, he stares and points at it. The right selection is finalized by user's correction through the graphical feedback on the LCD. It means that the feedback lessens user's confusion caused by the gap.

5. Experiment

We designed and conducted an experiment to verify the effectiveness of the proposed algorithm compared with classic ray-based selection technique. The experiment has been designed to evaluate the selection performance in terms of selection time.

5.1. Experimental setup

In order to adapt our adaptive angle assignment algorithm and verify it as well as to decide the threshold value, A_{TH}, we set up an experiment environment whose virtual space is represented in Figure 6. In the corresponding real space, 7 same sized LCD monitors (from D0 to D7, in Figure 6) were deployed at equal intervals. 13 males ranged in ages from 23 to 31 participated in this experiment and they were requested to make 70 correct selections, 10 times per each device, in a randomly generated order. The location of user is fixed at the designated point which was 180cm away from D0. We measured average time to select each device in both cases of using the ray-based technique and our algorithm.

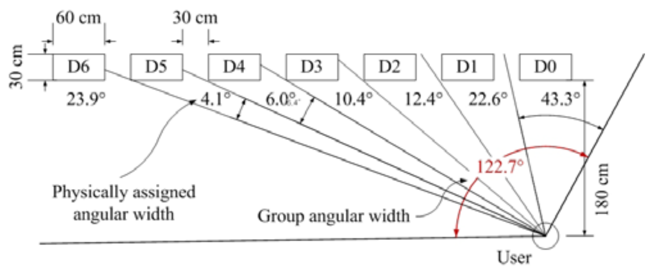
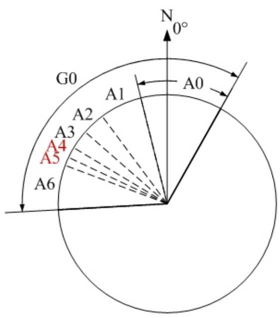
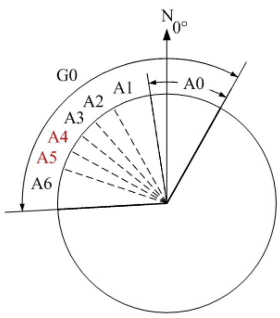


Figure 6. Virtual space of the experiment environment



	Start (deg)	End (deg)	Size (deg)
A0	330.50	13.80	43.30
A1	13.80	36.40	22.60
A2	36.40	48.80	12.40
A3	48.80	59.20	10.40
A4	59.20	65.20	6.00
A5	65.20	69.30	4.10
A6	69.30	93.20	23.90
G0	330.50	93.50	122.70

Figure 7. Angle table for the situation shown in Figure 6. D4 and D5 have relatively small angular widths.



	Start (deg)	End (deg)	Size (deg)
A0	330.50	8.530	38.03
A1	8.53	29.14	20.61
A2	29.14	41.16	12.02
A3	41.16	51.50	10.34
A4	51.50	61.50	10.00
A5	61.50	71.50	10.00
A6	71.50	93.20	21.70
G0	330.50	93.50	122.70

Figure 8. After the angle reassignment. Both D4 and D5 now have A_{TH} (10 deg).

5.2. Experimental results

Figure 6 and 7 show how the angular widths were physically assigned at this situation and Figure 8 shows the reassigned angular width according to our algorithm. Due to the effect of marginal angle, A_{mar}, which was set to 20 degrees in this experiment, both end devices (D0 and D6) were assigned larger angles than others, while the assigned angles of D4 and D5 were relatively small.

The result of the experiment is shown in Figure 9. Figure 9 shows the average time spend to select each device correctly. In case of using the ray-based technique, selection time significantly increased when the angular width is less than 10 degrees. From the result, we convinced that the selection action with *iThrow* follows Fitts' law and we decided a reasonable value of A_{TH} as 10 degrees. This value is a parameter determined by the user's experiences and can vary with the user characteristics.

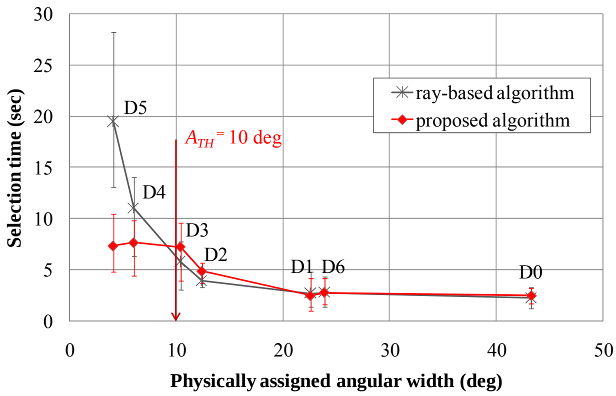


Figure 9. Effect of adaptive angle assignment on selection time.

The proposed algorithm prevented the rapid increase in selection time when a physically assigned angular width became lower than A_{TH}. Adaptive angle assignment reduced average time for selecting D5 whose physically assigned angle was 4.1 degrees about 62.6%. However, in case of D2 and D3, even though their angular widths were nearly unchanged, selections took slightly longer time than before. The reason is that a little gap between the original angular region and the reassigned angular region exists; the start and end degree of each angle may be changed. Unlike classic ray-based selection which doesn't have any gap between original and reassigned angular regions, the proposed algorithm requires some additional movements to compensate the gap. However, it does not affect the selection performance much because of general operating pattern of users as described in section 4.2.

6. Conclusions

We have described *iThrow* system and its selection algorithm. In our system, users can interact with others or public devices in user-friendly gestures. The proposed selection algorithm improves selection speed significantly.

As future works, *iThrow* system should be extended to 3 dimensional spaces and the optimal value of A_{TH}, which is a parameter determined by the user's experiences and characteristics, should be chosen for each individual users.

References

[1] J. Lee, S.-H. Lim, J.-W. Yoo, K.-W. Park, H.-J. Choi and K. H. Park, "A Ubiquitous Fashionable Computer with an *i-Throw* Device on a Location-based Service Environment," 2nd IEEE International Symposium on Pervasive Computer and Ad Hoc Communications, May 2007.

[2] M. Back, S. Lahlow, R. Ballagas, S. Letsithichai,M. Inagaki, K. Horikira and J. Huang, "Usable Ubiquitous Computing in Next-generation Conference Rooms: Design, Evaluation, and Architecture," UbiComp 2006 Workshop, Sep. 2006.

[3] Y. Song, S. Moon, G. Shim and D. Park, "Mu-ware: A Middleware Framework for Wearable Computer and Ubiquitous Computing Environment," Middleware Support for Pervasive Computing Workshop at the 5th Conf. on Pervasive Computing & Communications (PerCom 2007), pp. 455-460, Mar. 2007.

[4] A. Wilson and S. Shafer, "XWand: UI for Intelligent Spaces," Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 545-552, April, 2003.

[5] P. M. Fitts, "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," Journal of Experimental Psychology, vol. 47, pp. 381-391, 1954.

[6] G. V. Kondraske, "An Angular Motion Fitts' Law for Human Performance Modeling and Prediction," IEEE Engineering in Medicine and Biology Society, pp. 207-308, Nov. 1994.

[7] UPnP Forum, UPnP Device Architecture 1.0, Version 1.0.1, Dec. 2003.

[8] Ubisense, http://www.ubisense.net

[9] Freescale Semiconductor, 3-axes accelerometers, MMA7260Q, http://www.freescale.com/

[10] Honywell, Magneto-resistive sensor, HMC1053, http://www.ssec.honeywell.com/