

A Reactive Presentation Support System based on a Slide Object Manipulation Method

Hiroyuki Yamada, Shun Shiramatsu, Tadachika Ozono, Toramatsu Shintani
Dept. of Computer Science and Engineering
Graduate School of Engineering, Nagoya Institute of Technology
Email: {hyamada, siramatu, ozono, tora@toralab.org}

Abstract—We realized a system that provides manipulations of slide objects to enable users to give a reactive presentation. The “reactive presentation” refers to a presentation that can immediately react an unexpected context change. A presenter requires applying the presentation to the unexpected context such as the state of audiences, unexpected questions and audiences’ knowledge. Attention controls by animations and effects are effective to deal with the context. Animations and effects such as movements and changes of the size and color and feature of objects on slides can control audiences’ attention and communicate non-verbal information. However, the conventional PowerPoint did not support real-time manipulations of slide objects at the presentation mode. We expected that manipulations of objects on slides such as moving, zooming and adding effects enable presenters to control audiences’ attention according to the context change reactively. In this paper, we describe methods of the implementation of manipulating slide objects system with input devices properly, and discuss examinations of personalization to enhance operability of the system.

I. INTRODUCTION

A presenter requires applying the presentation to the unexpected context such as the state of audiences, unexpected questions and audiences’ knowledge. Attention controls by animations and effects are effective to deal with the context. However, the conventional PowerPoint did not support instant manipulations of slide objects at the presentation mode. In this paper, we describe methods of controlling audiences’ attention and adding visual information in the presentation.

We realized a system that provides manipulations of slide objects to enable users to give a reactive presentation. The “reactive presentation” refers to a presentation that can immediately react an unexpected context change. The key point of realizing methods for controlling audience’s attention during a presentation is developing functions for manipulating objects on slides in real time during a presentation.

Studies on presentation support systems have been conducted for long time; however, studies about the reactive presentation have never been conducted in the previous researches. Thomas et al., (1993) proposed a presentation support system to realize 16 gestures with a sensor globe to control presentation slides[1]. Shoemaker et al., (2007) proposed an interesting approach that enables pointing by a user’s shadow using perspective projection[2]. This system is an ingenious in that the system interacts with a user’s shadow. Our focus is to develop a presentation support system, which enables us to react unexpected context changes. We develop a new approach to properly react audience requests during presentations.

The problem is that existing presentation applications (e.g. Microsoft PowerPoint and Apple Keynote) do not support to edit of slide objects during the presentation mode. We cannot apply new animations and effects to slide objects when we need to immediately get attention during the presentation mode. On presentation applications, we can apply the abundant animations and effects to objects on slides. It is also beneficial for us to utilize the rich animations and effects under unexpected situations during presentations.

We propose a system for manipulating objects on slides in real time during a presentation to realize the reactive presentation. Manipulating slide objects such as moving, zooming and adding some effects, the presenter can guide and keep the audiences’ attention, and communicate information in visuo-spatial through representing the animations and effects of slide objects reactively. We explain how to implement a real-time slide object manipulation method and evaluate the usability of the method with a gesture input device in various situations.

This paper is organized as follows. In Section II, we describe what are essential functions to implement a system for reactive presentations. In Section III, we describe how to use various devices for manipulating slide objects properly and the design of the interface. In Section IV, we describe implementation methods of the system for manipulating slides objects. In Session V, we discuss the result of examinations of personalization to enhance operability of the system. In Section VI, we describe the conclusion.

II. REACTIVE PRESENTATION

We discuss how to realize the reactive presentation support system. The reactive presentation means controlling the attention of audiences by using movable slide objects under unexpected context changes during presentations without using the edit mode of presentation applications.

We define the reactive presentation as a presentation environment that a presenter can control attention by adding animations and effects to deal with certain context and reflecting in the presentation. Examples of the context include the state of audiences, unexpected questions and audiences’ knowledge. We show an example in case to use transitions of slide objects below. Transition is one of the animations to communicate movements of slide objects to audience. If a slide objects move to a goal and the route is changed by conditions, we need to set the route, speed and timing for each condition to move slide objects on slides before presentations. In case that an audience ask the presenter to indicate the route on unexpected

conditions that it is difficult to prepare before, the presenter can only explain by speaking or point.

A. Attention Control

Attention control in presentations is to guide and keep audiences' attention to crucial parts of slides it is effective to deal with the context in the presentation. Generally, the kinds of human load are classified into cognitive, visual, and exercise load in descending order of the size of the load. Looking for the corresponded part of the slide to what the presenter saying is classified as cognitive load. Providing relations between the presenter saying and the slide objects visually, we expect to increase the visual load on the one hand, but on the other hand decrease the cognitive load. The attention- guiding by means of cueing reduces extraneous cognitive load of the audience[3].

Exogenous attention can force to control person's attention unconsciously. To take an example, the audiences' attention is engaged in the movement and the variation of the color and size of slide objects by using animations unconsciously. On the other hand, endogenous attention is to pay attention consciously. For example, in the case of audiences look for and pay attention to the relevant parts of slides to what a presenter saying. Attention is the working to take only necessary information from too much information. Attention is controlled consciously or unconsciously by the attributes of the position, color, feature, etc. Exogenous attention and endogenous attention are responsible for deciding where to pay attention.

B. Animations for Attention Control

Using exogenous attention[4] by animations and effects is useful to guide the audiences' attention properly with the low load for the audiences. Guiding and keeping the attention visually by using animations and effects reduce the audiences' cognitive load[5]. We often use animations and effects such as highlight, transition, zooming for objects on the slides when we do the presentation by means of PCs and the projector. The benefits of animations are guiding the audiences' attention, keeping the attention and communicating non-verbal information.

Furthermore, Animations and effects can communicate non-verbal information to audiences effectively[6]; with expressing the variation of movements, color and features for slide objects. Pictures and Shapes represent visuo-spatial information that is high effective for human to understand such as feature and color and size. Animation adds the more visuo-spatial information such as the movement, transition and enhancing.

C. Slide Object Manipulation for Instant Attention Control

We describe what functions are necessary to implement a system for reactive presentations. We implement functions that a user can manipulate slide objects in the slide show for reactive presentations. The function is designed for manipulating objects on slides in real time during a presentation to realize the reactive presentation.

A user should manipulate slide objects in the slide show without changing to the edit mode to avoid losing audience

attention. Eye direction and attention are almost entirely consistent[7][8]. More salient cueing engages eye direction and attention more powerful. We should avoid the risk of carelessly attention. Presentation applications such as Microsoft PowerPoint and Apple Keynote provide abundant and wonderful animations and effects for presentations. However, Adding and editing animations and effects such as movements and changing the size and color of slide objects is possible only if the applications enable editing mode. Nevertheless, the slide show mode is better to concentrate audiences' attention at the time of presentation because there is no salient cueing with the exception of slide objects on the slides at the slide show mode.

Manipulating objects on slides in real time during a presentation to realize the reactive presentation. Manipulating slide objects such as moving, zooming and adding some effects, the presenter can guide and keep the audiences' attention, and communicate information in visuo-spatial through representing the animations and effects of slide objects reactively. For example, if we are able to manipulate slide objects in presentation, we can re-edit and add animations and effects for the context that is difficult to prepare before.

III. METHODS FOR MANIPULATING SLIDE OBJECTS

We discuss about various methods to manipulate slide objects. Presently typical used input devices are Mouse and Touchscreen. Recently Kinect and Leap Motion that are motion sensors communicate raw position information of hands and fingers to computer appeared at high precision and low price. These input devices are also able to communicate the information of click, tap and human gesture. In this section, we suggest methods and environments to manipulate slide objects by input devices that are used on difference scenes and purposes properly.

A. Properly using input devices for manipulating slide objects

Evaluating the mouse, touchscreen, Leap Motion and Kinect from the viewpoint of usability, intuitively and user mobility, we suggest methods for manipulating slide objects by using these devices that are used on difference scenes and purposes. When manipulating slide objects, high usability means a user can execute intended operations easily. Intuitively means a user how to associate manipulation methods with gestures and the actual manipulation methods[9]. High user mobility means a user is not limited to the position to manipulate slide objects. Thus, the recognizable range of the sensor is wide.

The most common input device is the mouse. Mouse has high usability in terms of its operation methods that are well known. However, in the case where all manipulations of slide objects defined in this research are satisfied, the manipulations with the mouse become complicated and low of usability. For this reason, using the mouse together with Keyboard, we construct a precise and usability-prioritized system to manipulate slide objects.

Touchscreen has high usability and intuitiveness through fingertip touch with which inputting the locations to computer correspond directly. Additionally, if the tablet PC supports the projection with a projector via Wi-Fi, the user can interact by moving in the area in which a tablet PC can communicate. That

is, in case of using Touchscreen, we can construct a system with usability, intuitiveness, and user mobility to manipulate slide objects.

Tsukada et al., (2002) realized the gesture input interface using a user's fingers with sensor gloves. Adopting movement of dynamic gesture into a presentation, they provide to concentrate audiences' attention by visual impacts and presence[10]. In this research, we also realize the gesture input interface with motion sensors. Motion sensor supports tracking locations of hands, fingers and skeletons in 3D, hence they are expressive for manipulations and can create triggers that users can understand more easily. Leap Motion supports a high precise finger tracking. By using Leap Motion, we can construct a highly intuitive interface. Kinect has high user mobility that supports skeletons tracking in the wide recognizable area. By using Kinect, we can construct the interface with high user mobility and interact through the movement of users' whole body.

From the above evaluation, in case a user needs precision and usability in priority, Mouse together with Keyboard are suitable for manipulating slide objects. In case where a user needs usability, intuitiveness and optionally mobility, Touchscreen is suitable. In case where the user require reducing the load of interaction, Leap Motion that supports intuitive manipulation by highly precise finger detection is suitable. In case the movement of presenter is involved in the presentation, Kinect is suitable due to support high user mobility and detecting user's skeletons in wide recognizable area.

B. Interface Design

Thinking about the interface design to manipulate slide objects in the presentation is important for usability and reducing user load. Especially, the design of gestures for triggers of manipulations by using devices that support multiple pointing such as Touchscreen, Leap Motion and Kinect directly affect usability, intuitiveness and user mobility. Gesture is the movement of a body with functional meaning in computer science[11]. There are three benefits to use the gesture input. First, gesture inputs are high expressiveness for this reason that gesture input interface can set a lot of simple triggers which the user understand easier by extracting the movements themselves of user's fingers, hands and skeletons. Second, the user doesn't need to be conscious of using input devices. Gesture input interface almost uses mounted or stationary sensors, that is, the user just have to think to input gestures. Third, gesture input interface is intuitive[12][13].

However, the gesture input interface has three defects. First, doing gestures continuously in the air causes that the user will be tired. Second, the motion sensors such as Leap Motion and Kinect especially take care of this defects that the user cannot gain physical feedbacks in comparisons with mouse clicks keyboard taps, etc. Third, the user needs to learn the gestures for controlling a computer.

When designing the gesture input interface, it must be assessed its defects. The gesture input interface to manipulate slide objects should support the gestures bound to mundane actions such as the action of grasping something a person would hold, and the gestures are widely known such as Swipe and Pinch-in/out. For the user using Leap Motion and Kinect,

it is also significant to feed back the result of manipulations to the user visually because there is no clear feedback in comparison with Mouse and Touchscreen, which can feed back physically such as click and tap.

C. Gesture Input Processing

Algorithm 1 Gesture Input Processing Algorithm

```

1: while true do
2:    $\langle M, P, r \rangle \leftarrow WaitEvent()$ 
3:   if  $r > r_{prev}$  then
4:      $ResetState()$ 
5:      $r_{prev} \leftarrow r$ 
6:   else if  $r < r_{prev}$  then
7:     continue
8:   end if
9:    $G \leftarrow recognizeGestures(M, P)$ 
10:   $P_s \leftarrow convertToScreen(P)$ 
11:   $P'_s \leftarrow stabilizedPosition(P_s)$ 
12:   $c \leftarrow determineControlPoint(P)$ 
13:   $c_s \leftarrow convertToScreen(c)$ 
14:   $\langle x'_{cs}, y'_{cs} \rangle \leftarrow stabilizedPosition(c_s)$ 
15:   $moveMouseCursor(x'_{cs}, y'_{cs})$ 
16:   $o \leftarrow findSlideObjectAt(x'_{cs}, y'_{cs})$ 
17:   $m \leftarrow detectMode(M, G, P)$ 
18:  if  $m$  is Gripping AND MoveXY then
19:     $moveSlideObjectTo(x'_{cs}, y'_{cs}, o)$ 
20:  else if  $m$  is Gripping AND MoveZ then
21:     $changeZIndex(c, o)$ 
22:  else if  $m$  is Pinch-in/out then
23:     $changeSizeOfSlideObject(c_s, o)$ 
24:  else if  $m$  is others then
25:    do processing of pointer, pen and swipe
26:  else
27:    do the default actions
28:  end if
29:   $r_{prev} \leftarrow 0$ 
30: end while

```

Algorithm 1 is to process gesture inputs for manipulating slide objects with multiple pointing devices such as Touchscreen, Leap Motion and Kinect. At line 2, when the events are published from input devices, getting information of events consists of $\langle M, P, r \rangle$. M is the set of flags. The flags indicate whether Gripping, MoveXY, MoveZ, Pinch-in/out, Pointer, Pen or Swipe mode is turned on or off. When gripping mode is turned on, a user is about to grab a slide object. MoveXY and MoveZ mode are to move a slide object. MoveZ mode is only used when the devices capable of acquiring 3D position values such as Leap Motion and Kinect. In front of Leap Motion or Kinect, X is the horizontal component, Y is the vertical component and Z is the depth component. Pinch-in/out mode is to zoom, Pointer mode is to display a pointer and Swipe mode is for page transitions. $P = (p_1, p_2, \dots, p_n)$ is the array of detected pointing (n is the number of detected pointing). Pointing p has a tuple $p : \langle x, y \rangle$ or $p : \langle x, y, z \rangle$ representing the positional information on 2D or 3D. r is the priority that is assigned for each input device. At lines 3 through 8, we determine a valid event from plural occurred events based on priority of each device. At line 9, get the array of gestures $G = (g_1, g_2, \dots, g_n)$ from M, P (n is the number of detected

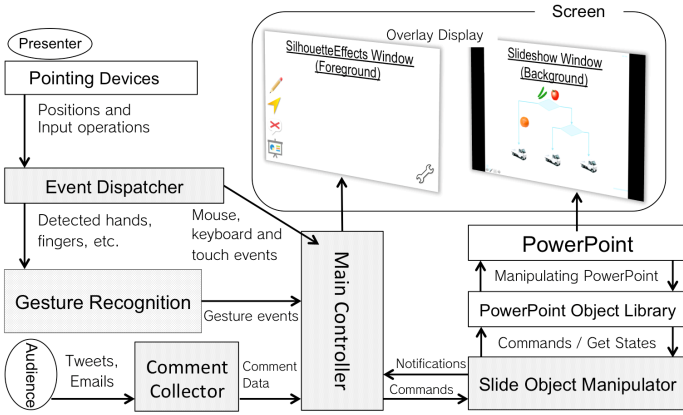


Fig. 1. System diagram.

gestures). Gesture g consists of $g : \langle t, s, j \rangle$ where t is the kind of gesture, s is the state of gesture and j is the time of the gesture was detected. At line 10, convert the obtained positional information to P_s based on a screen coordinate. At line 11, get the stabilized positions P'_s .

At line 12, determine an origin point c for manipulating. c is the position of the palm of the detected hand on Leap Motion and Kinect. Displayed a mouse cursor on a screen follows the origin point. At lines 13 through 14, calculate converted positional information to the screen coordinate c_s and the stabilized position $\langle x'_{cs}, y'_{cs} \rangle$. At line 16, do hit testing between the origin point and the slide object. If the mouse cursor is in the slide object, get the information of the slide object. At line 17, determine the current mode m from M, G, P . If m is Gripping and MoveXY, move the slide object o to $\langle x'_{cs}, y'_{cs} \rangle$ at line 19. If m is Gripping and MoveZ, exchange the overlapping sequence orders of slide objects at line 21. If m is Pinch-in/out, change the size of the slide object at line 23. If m is Pointer, move the displayed pointer to $\langle x'_{cs}, y'_{cs} \rangle$. If m is Pen, paint on the basis of the position $\langle x'_{cs}, y'_{cs} \rangle$. If m is Swipe, determine the direction of the swipe from G and transit pages.

IV. IMPLEMENTATION

We implemented a system for manipulations of slide objects called SilhouetteEffects. The system supports Microsoft PowerPoint 2010 and 2013. As for the development environment, we use C# on .NET Framework with Visual Studio 2013. We also use LeapMotion Developer SDK¹ and Kinect SDK² to control the motion sensors. As for controlling PowerPoint, we use PowerPoint Object Library³. We describes the structure and the functions of our system in this section.

Our approach can also be applied to distant presentation support by using a remote synchronization mechanism. We focus on slide object manipulation for a single user due to limitations of space.

A. System Structure

Figure 1 shows the system diagram. The system is roughly divided into three parts. The first is the sensor part that detects user inputs with the device that is Mouse, Keyboard, Touchscreen, Leap Motion or Kinect. The second is the software part that processes some information obtained from the sensor part. The third is the screen part that reflects the results of processing on the software part. The software part has six components. These components are for dispatching events, detecting gestures from user inputs, manipulating the slide objects, collecting audiences' comments, the main controller and PowerPoint. The screen part has two layers. The foreground layer displays the control menu of the system. The background layer displays the slide show of PowerPoint.

When a presenter inputs a mouse click, drag and drop, touch, tap, pinch-in/out, a movement of the fingers, hands and skeletons, the sensor part catches these inputs and sends them to the event dispatcher component. This component differentiates between mouse or keyboard inputs and gesture inputs that are the information of the movements of fingers, hands and skeletons. Mouse or keyboard inputs are dispatched as mouse events or keyboard events to the main controller. Gesture inputs are dispatched to the detecting gesture component. This component understands the types of gestures with the **Algorithm 1**, and sends the types to the main controller. The component collecting audiences' comments is always listening related tweets on Twitter and mails from audiences during the presentation. When the component catches the comments, the data of comments are sent to the main controller. The data are consisted of the name of the comment author, the timestamp when the comment was sent and the content of the comment. The main controller manages dispatching of the commands in accordance with the current mode to the component controlling slide objects. The main controller also manages to display each current state of mode and the comments.

The component to manipulate slide objects interprets the main controller's commands, as that PowerPoint Object Library (PPOL) can understand it. PPOL is a Library to manipulate PowerPoint such as opening PowerPoint files, creating slides and manipulating the slide show. However, PPOL cannot understand the commands that PowerPoint cannot do such as detecting mouse over the slide objects and manipulate slide objects smoothly on the slide show. The component to manipulate the slide objects plays the role of wrapper to make PPOL understand the commands. This component also sends the information that is the current position of the slide show, the number of objects and the types and whether there are any animations and effects on the slides to the main controller. This system has the authority to control PowerPoint generally. However, this system is equipped to transfer the authority to PowerPoint. In this case, the inputs from a mouse or keyboard send to PowerPoint directly without sending via the event dispatcher component.

B. Function

This subsection describes how to manipulate the slide objects in this system. Figure 2 shows the operations supported by the proposed system. The recognizable gestures with multiple pointing devices that are Touchscreen, Leap Motion and

¹<https://developer.leapmotion.com>

²<http://www.microsoft.com/en-us/kinectforwindows/>

³<http://msdn.microsoft.com/en-us/library/office/>





















Mouse & Keyboard	Touchscreen	Kinect	LeapMotion	Manipulating Slide Object
Click & Drag 	Pan 	One-handed Grip 	One-handed Grip 	Move
Shift + 	Pinch-in/out 	Two-handed Grip 	Pinch-in/out 	Zoom
NaN	Double Tap 	One-handed Push 	Handle Pen or A Finger Protruded forward 	Display Pointer
Left Arrow Key 	Left Swipe 	Left Swipe 	Left Swipe 	Go previous slide
Right Arrow Key 	Right Swipe 	Right Swipe 	Right Swipe 	Go next slide
NaN	NaN	NaN	Triple key taps 	Write memo

Fig. 2. Operations supported by the proposed system.

Kinect be closed to users are likely to associate gestures when they manipulate slide objects. This means that users readily understand the correspondence of the recognizable gestures to execute operations more easily.

With Leap Motion, the position of the cursor or pointer on the screen with the center of a palm position. If you want to manipulate a slide object, you input the gesture corresponding to the manipulation. In case of moving a slide object, the user does the action of grabbing the subjected slide object. The user can move it until release of the slide object. In this system, we define grabbing something as Grip and releasing it as GripRelease i.e. the action of extending the user's fingers. Slide objects can be moved up and down, left and right, and in front and behind the slides. In case of moving in front and behind, the user can change the stacking order of the slide objects. Moving the user's hand along the y-axis, the user can move up and down a slide object. Moving the user's hand along x-axis, the user can move left and right. Moving the user's hand along z-axis, the user can move in front and behind. When zooming a slide object, the user does the gesture of pinch-in/out. In case of displaying a pointer, the user sticks the user's finger further in than above the Leap Motion. To go back the previous slide, the user does the gesture of SwipeLeft. To go the next slide, the user does the gesture of SwipeRight. These gestures bound to mundane actions. Regarding the case of writing memos, the user can write memos after mimicking tapping a key three times in the air. This gesture is not associated with writing memo because of the purpose of this binding is to prevent malfunctions for other operations.

As for the manipulating methods with Kinect, there are five recognizable gestures. The number of recognizable gestures is less than with Leap Motion. We define five gestures as Grip, GripRelease, Push, SwipeLeft, and SwipeRight. Each gesture is corresponded to following operations: start to move a slide object, end to move a slide object, display a pointer, go the next slide, and go back the previous slide. Grip is the gesture of gripping in the air. GripRelease is the gesture of extending fingers. Push is the gesture of placing the user's hand in front of the body. Moving the user's hand left and right correspond to SwipeLeft and SwipeRight.

Using only the mouse is not sufficient to manipulate a slide objects like multiple pointing devices. In this system, combining the mouse with the Keyboard, the user can do the same manipulation as them.

For supporting communications between a presenter and audiences, this system also has the function of collecting audiences' comments in presentations with Twitter and E-mail. By inserting the specified hash tag inside tweets or sending mail to a specific e-mail address, the audiences can post comments. Collected comments are displayed on the left side of the screen. Embedded comments in slides by PowerPoint are also displayed. The comments have been displayed until the current slide is transited. When PowerPoint is closed, the posted comments are embedded within each slide. This function provides the audiences with the ability to ask questions and comments during the presentation, and the presenter with the ability to log the comments after the presentation.

V. EVALUATION AND DISCUSSION

In this section, we discuss the investigation of a correlation between operability and the size of users' hand to the moving slide objects with Leap Motion.

As opinions of manipulating slide objects on our system with Leap Motion, one user said that it is easy to use and the other user said that it is hard to use. These opinions indicate that each user feels different usability and intuitively. We personalize our system for each user. Personalization is a method for enhancing usability and intuitively on gesture input interface, and besides, to optimize a system for a user's characteristics. The characteristics on the interface by means of Leap Motion are the size and feature of the hand, the user's habitual gestures, etc. we focused on the size of the hand and expected that the feeling of usability and intuitively depend on the size of the user's hand. Doing the action of grasping something a person would hold after moving the mouse cursor onto a slide object, a user can determine which slide objects to manipulate. The threshold to discriminate whether or not the user grabbed a slide object is the radius of the formed sphere by detected user's fingers. The user moves the cursor

Fixed Threshold (62mm)	A	B	C	D	E
Hand Size [cm]	18.5	18.5	20.0	17.5	17.0
Average of Completion Time [sec]	48.4	43.1	38.1	46.0	59.8
SD of Completion Time	8.1	3.9	7.2	2.9	16.6
Threshold / Hand Size	3.351	3.351	3.100	3.543	3.647
Free Threshold	A	B	C	D	E
Hand Size [cm]	18.5	18.5	20.0	17.5	17.0
Set Threshold [mm]	70.0	60.0	62.0	60.0	62.0
Average of Completion Time [sec]	28.0	29.8	37.0	37.1	29.4
SD of Completion Time	5.7	4.2	14.1	8.8	4.1
Threshold / Hand Size	3.784	3.243	3.100	3.429	3.647

Fig. 3. Experiment result.

following the center position of the palm onto a slide object, and grabs the slide object, in other words, makes the radius of the sphere smaller. If the radius bellows the set threshold, the system recognize as the user grabbing the slide object. Therefore, we expected that the easiness of manipulating slide objects depends on the set threshold which has a correlation to the size of the hand.

In order to examine the expectation, we carried out to an experiment that five subjects (A-E) who are members of our laboratory move slide objects with Leap Motion. In this experiment, we measured the completion time to complete a task that is moving slide objects from the initial place to the specified place. Five subjects have the same proficiency. The subjects perform the task three times in each case that the threshold is fixed to 62mm and the subject can set the threshold freely (six times in total). The subjects set the threshold that is most easily for them to manipulate slide objects. The size of the hand is a length from the wrist to the middle fingertip.

The result is shown in Figure 3. Fitting the threshold for each user is effective because in case that the subjects could set the threshold freely, they completed the tasks earlier than in case that the threshold was fixed. Figure 4 shows a relationship between completion times and the ratio of threshold and the size of the hand. The horizontal axis represents the ratio of the threshold and the size of the hand. The vertical axis represents time to complete the task. If there exists an optimal ratio, a downward peak should appear in the graph. It is, however, difficult to find a downward peak and to estimate the optimal ratio of threshold and the size of the hand from this result. It is assumed that the differences from associated gestures by users have bigger influence than the size of the hand. The result indicates that personalization based on machine learning techniques is needed to improve the accuracy and to enhance the usability and intuitiveness on gesture input interface.

VI. CONCLUSION

We realized the system for reactive presentations that a presenter can deal with the context that is difficult to prepare before. The reactive presentation is controlling the attention of audiences by using movable slide objects under unexpected context changes during presentations without using the edit mode of presentation applications. The system provides manipulations of slide objects such as moving, zooming and adding effects in the slide show. On the design for gesture input interface, we took care of the gestures are bind to the routinely

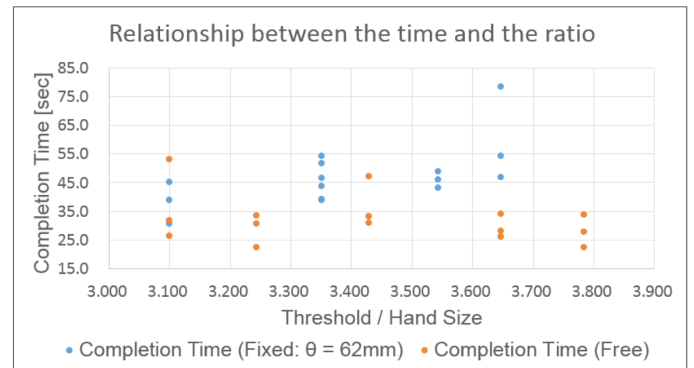


Fig. 4. The task completion time and the ratio of threshold and hand size.

actions to enhance usability and intuitively of the system on gesture input interface.

We evaluated our system on some devices from the viewpoints of usability, intuitively and user mobility, and suggested uses of this system with properly input devices. From the result of examinations indicates that there is no correlation between the size of a user's hand and operability, the system learning the user's unique gestures for personalization is effective to enhance usability and intuitiveness.

REFERENCES

- [1] T. Baudel, M. Beaudouin-Lafon, *Charade: remote control of objects using free-hand gestures*, Communications of the ACM - Special issue on computer augmented environments: back to the real world, Vol. 36, Issue. 7, 1993.
- [2] S. Garth, T. Anthony and B. Kellogg, *Shadow reaching: a new perspective on interaction for large displays*, In UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology, pp. 53–56, 2007.
- [3] F. Amadiou, C. Marine and C. Laimay, *The attention-guiding effect and cognitive load in the comprehension of animations*, Computers in Human Behavior, Vol. 27 pp. 36–40, 2011.
- [4] H. Pashlar, C. Johnston and E. Ruthruff, *Attention and performance*, *Annu Rev psychol*, Vol. 52, pp. 629–651, 2001.
- [5] J. Maria and M. Ferreira, *Intelligent classrooms and smart software: Teaching and learning in today's university*, Education and Information Technologies, Volume. 17, Issue. 1, pp. 3–25, 2012.
- [6] K. Meyer, T. Rasch and W. Schnotz, *Effects of animaton's speed of presentation on perceptual processing and learning*, Learning and Instruction, Vol. 20, Issue. 2, pp. 136–145, 2010.
- [7] B. Khurana and E. Kowler, *Shared attentional control of smooth eye movements and perception*, Vision Research 27, pp. 1603–1618, 1987.
- [8] E. Kowler, E. Anderson, B. Doshier and E. Blaser, *The role of attention in the programming of saccades*, Vision Res 35, pp. 1897–1916, 1995.
- [9] B. Moniruzzaman, and P. Rich, *Gesture-controlled user interfaces, what have we done and what's next?*, Proc. Fifth Collaborative Research Symposium on SEIB, pp. 59–60, 2009.
- [10] K. Tsukada and M. Yasumura, *Ubi-Finger: Gesture Input Device for Mobile Use*, Proceedings of APCHI 2002, Vol. 1, pp. 388–400, 2002.
- [11] G. Kurtenbach and E. Hultheen, *Gesture-Based Communication in Human-Computer Interaction*, The Art of Human Computer Interface Design, pp. 309–317, 1990.
- [12] V. Radu-Daniel, *Interfaces That Should Feel Right: Natural Interaction with Multimedia In-formation*, Recent Advances in Multimedia Signal Processing and Communications Studies in Computational Intelligence, Vol. 231, pp. 145–170.
- [13] V. Radu-Daniel, *On designing interactivity awareness for ambient displays*, Multimedia Tools and Applications, Vol. 66, Issue. 1, pp. 59–80, 2013.