Presentation Interface Based on Gesture and Voice Recognition

Jinuk Kim, Sehoon Kim, Kwangjin Hong, David Jean, and Keechul Jung

School of Media, Soong-sil University, Seoul, South Korea
{jinuk,sehoon,hongmsz,jeandavid,kcjung}@ssu.ac.kr

Abstract. In this paper, we introduce a Kinect based interface that recognizes gestures and voice. We have developed an interface to control presentations such as speeches or lectures. It is possible to receive the coordinates of the body, and recognize gestures and positions of the hand. Data received by the camera in Kinect are used to create a hook between the user hand and a presentation application such as Microsoft Powerpoint. Our interface is able to recognize grip and push gestures from the presenter. The result of this gesture recognition generates a signal to the presentation application, such as shortcuts to change slides or make use of additional tools. It is also possible to start and end the presentation by voice using our voice recognition tool. Additionally we show some tools that not only change the slides, but also provide more options to the presenter such as memo tools to directly highlight some parts of a slide, and even an eraser. This paper describes all the methodology and presents the result of our tests session. We are effectively able to improve the presentation capability of the presenter and think that such interface can be commercialized for presentation and other type of use.

Keywords: Gesture recognition, Voice recognition, Presentation interface.

1 Introduction

After 1980, generalized GUI interfaces consist of restricted environments that only use mouse. But the development of input and output devices pushed interaction from a computer centric paradigm toward a more human and natural nature. MIT media lab started graphic interface development research with Chris Schmandt's voice and gesture in 1979[1]. This project has attracted more attention on gesture recognition as an alternative for eliminating the inconvenience of using input devices in specific contexts.

Unlike basic typing and mouse control, it becomes necessary to explore natural interaction between user and system for game and contents. In such areas, interactive systems consist of spaces and sensors that generate interactive and immersing experiences. It is for example used on visitors moving in a museum that could see some specific content that pop-up just by moving or using some gestures. We believe that it would create a better experience.

James J. (Jong Hyuk) Park et al. (eds.), *Multimedia and Ubiquitous Engineering*,
Lecture Notes in Electrical Engineering 308,
DOI: 10.1007/978-3-642-54900-7_11, © Springer-Verlag Berlin Heidelberg 2014

The development of interaction between computers and human can provide a lot of help in the everyday tasks, for example, in the case of presentations. When giving a presentation, natural behavior from the speaker is an important aspect that heightens the audience's concentration. When controlling a presentation using a device, these actions create an unnatural pause that can decrease the concentration of the audience.

This paper explores how to control a Powerpoint project by using Kinect sensor's skeleton position recognition in order to track hand position and recognize specific gestures. We use hand position to control the slides in Powerpoint while progressing the presentation naturally. We also developed a memo function that allows the user to write any content directly on the slide that is shown in Powerpoint. The user can start a presentation by using its voice and end it the same way. While doing the presentation, it is possible to move to another slide naturally. The presenter can improve its presentation capability by using a better interface environment.

2 Related Research

Especially in the fields related to computer vision, 2010 has seen a large number of studies based on movement and speech recognition, and still today, a lot are in progress. Kinect allows many researches to easily use a human skeleton, and such tool has generated many commercial applications.

Osunkoya[2] used the skeleton in order to obtain the coordinates of the hands and other specific parts (such as head or shoulders). The distances or respective positions of these different parts are then used to send events such has mouse move, or right and left clicks. These actions are then used to control Powerpoint. This method requires the users to place their hands at uncomfortable positions such as above their head, to insure the system can easily recognize the positions and send the right signals. Also, when moving from the position of one action to the position that recognizes another action, the risks to trigger another undesired action in between are high.

Gabacia[3] has developed a system that allows using simultaneously Kinect and Wiimote in order to control the mouse cursor during a presentation. This system allows the presenter to use voice recognition, Wiimote acceleration sensor and some buttons. Although this method uses Kinect and Wiimote, the control of the movement of the cursor is mainly based on the Wiimote device and doesn't make use of gesture recognition.

Another research proposed by Ren focuses on recognizing hand gestures using Kinect RGB camera to obtain the hands coordinates[4]. It is possible to recover the outline of the hand and make a timed series of the entire outline on each image to extract some feature vectors. This method shows 90% accuracy. However, the user must wear a black band on his hand to help the system reading the hand position correctly. Also, this method needs to compare all the raw data to all the features that are stored in a library, which is very computationally expensive.

Codeplex[5] is a system that uses the user body in order to control a slideshow. In Codeplex, the distance between the head and the guiding hand is constantly measured, and is used to trigger some functions. However, the lack of clear start and end of the gestures makes it difficult and causes many undesired actions. In order to fix this type of error, we believe that the beginning and end of the gestures must be clearly caught by the system.

Also, SmartTV proposes a personalized service that is based on gesture recognition. It is possible to control the latest generation of Samsung smart televisions with the new technology named Smart Interaction which actively make use of voice and hand gesture recognitions[6]. Recognition of the gestures uses the coordinate's values of the moving hands when they are in a grip posture. The main problems come as the system recognizes gestures that are not intended by the user, and process and undesired action.

Lately, lots of systems include more and more voice recognition in addition to gesture recognition, for researches as well as for commercial purposes. Apple's Siri[7], or Samsung's SVoice[8] offer services that make use of high level of technology in voice recognition.

3 System Structure and Functions

Kinect for Xbox 360 uses an input that consists of an RGB camera, an IR sensor, a microphone and a Tilt Motor. IR sensor can measure object's depth different from camera. Recognition range of Kinect is from 0.8m to 4m. Up and down movement of Kinect are recognized using tilt motor and microphone can recognize voice. Using Microsoft Kinect SDK v1.7 library it is possible to extract voice and skeleton information. It allows to recognizing gestures through skeleton information X, Y, Z coordinates. Microsoft Powerpoint slide show can be controlled by using mouse and keyboard. In this paper, we use Kinect to obtain user's body coordinates. Using the Kinect, we are able to recognize the position and gesture using coordinate values of a part of the body of the user. Especially, basic slide movement as well as memo, erase can be controlled.

Detected coordinates with the Kinect library are measured on the pixels of the Kinect common screen space. For this project, we have to manage 2 different spaces: one we call the application space and another we call the Kinect space. We first extract the coordinates of the hand from the Kinect skeleton space (K_x, K_y) and convert it to obtain the coordinates inside our application space (x, y). We are then able to process the movement.

However, these spaces have different coordinates systems and base units. The Kinect skeleton space has its base(0,0 coordinate) at the center of the Kinect view, in which the skeleton is displayed, and the unit used is meter. This space width and height are respectively called K_w and K_h. But the application space, has its base at the top left part of the screen, and the basic unit is pixel. Application space width and height are respectively called S_w and S_h. So we have to convert the points from one space to the other. The x and y coordinates that we need are extracted following the equation.

$$\begin{bmatrix} x \\ \mathrm{y} \end{bmatrix} = \begin{bmatrix} S_w/K_w & 0 & -S_w/2 \\ 0 & S_h/K_h & -S_h/2 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} K_x \\ K_y \\ 1 \end{bmatrix} \tag{1}$$

Kinect camera uses bilateral symmetry, like when a person looks at mirror. We set the Kinect in front of user and a screen is backside of the user. In this situation user moves the right hand and the cursor moves to left in the screen.

It means that the mouse controls is inverse to the speaker. To match a movement of the speaker in Kinect camera's coordinates, we extract the coordinates of the cursor (C$_{x,}$ C$_{y}$).We use equation2 to change the coordinates.

$$\begin{bmatrix} C_x \\ C_Y \end{bmatrix} = \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix} * \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} S_w \\ 0 \end{bmatrix} \tag{2}$$

We extract the coordinates value of pixel unit and we can control the mouse cursor to follow the right hand's x, y coordinates. We also developed functions that recognize user special hand movements to go to previous slide or next slide. Editing function is divided into two parts, one is memo function and the other is eraser function.

Each function specified by the user's gestures is recognized at the same time internally, and the corresponding shortcut is then executed. In a slideshow situation the user can invoke functions such as going to next slide, previous slide, staring the pen tool or eraser tool, drag event and click event.

We designed specific gestures for each of the functions we developed. For example, the grip gesture followed by a sweep to the right allows the user to display the next slide of the presentation. The following Table 2 shows all the functions and their corresponding gestures

Table 1. Gesture for controlling function

Function	Gesture
Mouse Click	Grip gesture
Mouse Click Release	Release gesture
Next Slide	Grip & sweep right
Back Slide	Grip & sweep left
Mouse Cursor Change	Press gesture

Move to the next slide is performed when right hand is in gripping state and release grip after it moves to the right. The move to the right is recognized when the position of the hand on the following frame shows a significant difference in X coordinate to the current position. The same process is used for the sweep to the left, however, we use the left hand to recognize it, and the X position is the opposite than its right counterpart. For the push and grip gestures, we are doing a similar process but using the Z coordinate this time. Microsoft Kinect SDK does not recognize the grip posture of a hand at a satisfying rate. It often happens that a grip posture is recognized although it was not intended by the user. It is especially true when the hand overlap with the body in the camera image. This is why we developed a more robust algorithm that recognizes the grip posture only when the user keeps is hand closed for 10 consecutive frames.

Our program uses another thread to process the voice recognition. We implemented in a multitasking way so that updates are done in real time, and it is possible to use other functions during the execution of the voice recognition. Voice recognition and gesture recognition use same Microsoft Kinect SDK, but internally voice recognition uses Speech Recognition Language pack(en-US) of the Microsoft Speech Library[9]. The voice recognition system searches inside the Speech Recognition Language pack(en-US) engine for specific words that we registered, and check the similarities with the words coming from the user. Words that are registered during the operation are recognized using the voice input form the microphone integrated in Kinect.

Before saying “Start Presentation” Kinect doesn’t recognizes any movement. After voice is recognized, the slideshow and the gesture recognition start. On the contrary, when saying “End presentation” the slideshow and gesture recognition end. Gesture and voice recognitions can be efficient ways to control a presentation through user’s gesture by using Kinect. However, we have to make sure that the Powerpoint application is the application running at the foreground insuring the shortcuts generated are passed to this application. For this, our interface uses hooking to control the mouse from received coordinate value from Kinect.

In computer programming, the term hooking covers a range of techniques used to alter or augment the behavior of an operation system, of applications, or of other software components by intercepting function calls or messages or events passed between software components. Code that handles such intercepted function calls, events or messages is called a ‘hooking’[10].

Use hooking, Powerpoint can be controlled by Kinect that recognize gesture like basic window mouse event handler.

4 Experimental Results

Our system experiments have been done in a basic lecture room. Distance between the Kinect and the user is 3.5m in the experimental space, and the range of recognition corresponds to the size of the image displayed by the projector for the presentation.

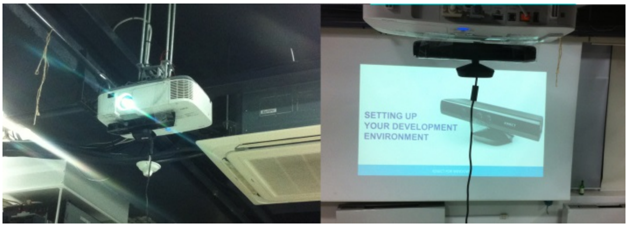

Fig. 1. Experimental space

In the experiment, we can confirm the start and end function of slide show is controlled by voice recognition, the sliding movements that control the next and previous slide, the movements to switch functions, the memo function and eraser function are all controlled using the gesture recognition integrated in our system.

Assuming a situation where the user has a presentation to do, he can starts this presentation by saying "start presentation". While the presentation progresses and the user wants to display the next slide, he puts his hand in a grip posture, moves it from left to right then release his hand posture to finish the gesture. Also, when we want to go back to the previous slide, the same gesture using the left hand and opposite direction is available (Fig. 2).

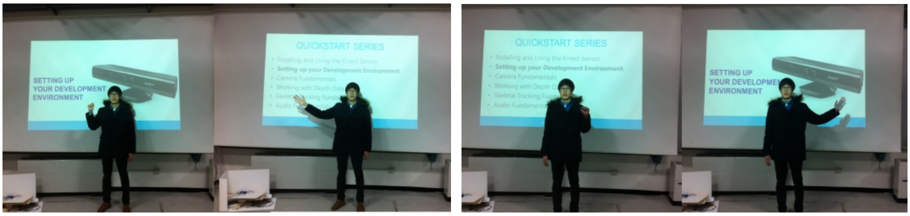
SETTING UP YOUR DEVELOPMENT ENVIRONMENT
QUICKSTART SERIES
QUICKSTART SERIES
SETTING UP YOUR DEVELOPMENT ENVIRONMENT

Fig. 2. Move next Slide & Move previous slide

In addition to the function to change the slides, our system provides functions in order to edit the slide on the fly, such as memo function and erase function. To use them, we must change the cursor directly in the Powerpoint application. To do so, we create gesture to switch from one function to another. When we are in the mode to change the display of the slides, we can pass to the next mode (memo tool) by using a gesture. Similarly we can pass from memo tool to eraser tool, or from eraser tool back to change slide tool. This is the switching function of interface(Fig. 3).

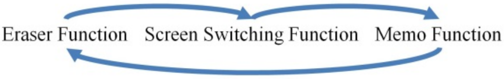

Fig. 3. Switching function root

The memo function is based on the coordinates of the hand. Memo directly modifies the slide. However, if we record all the hand gestures, we cannot obtain exactly what the user wants to draw, because the hand is constantly moving. So we use the grip posture to ease the drawing process. When the user draws with the memo tool, it just has to adopt the grip posture and draw. When the hand is released, the movement of the hands is free and no drawing is made. The same process is used for the eraser tool. When the user is on the eraser function state, it can just grip the hand and move it to erase the content of the slide, and then release its hand to stop erasing.

We chose to use the grip position of the hand to make sure we only consider what the user really want to do, and reduce the errors that are made with undesired gestures. The system is easy to use, since the user just has to change to the desired state of the function he wants to use, then grip his hand to do the gestures(Fig. 4).

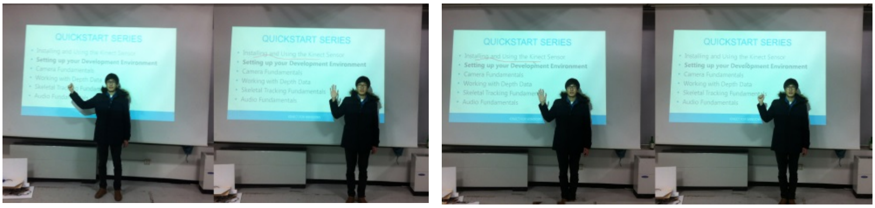

Fig. 4. Memo function & Eraser function

To finish the presentation, the user says "End Presentation". This automatically ends the slide show and the system stops the gestures recognition functions.

5 Conclusion

We introduced an application that makes it possible to start or end the slide show by using the voice recognition through a microphone Kinect. Also, by converting the skeleton image of someone with the Kinect's camera, it is possible to control the presentation. Unlike other researches that just allow to change slides, we added some functions such as a memo tool and an eraser tool. These functions augment the presenter capabilities a lot. Unlike previous studies which were only using the sweep movement, our application allows to use sweep and grip at the same time, and are used to effectively prevent undesired operations from the presenter. It is expected that the voice recognition module and gesture recognition module are used to control a presentation, but they may be re-used for other studies in multiple context, and will be of great help in future studies.

References

1. Bolt, R.A.: Put-that-there: Voice and gesture at the graphics interface, vol. 14(3). ACM (1980)
2. Osunkoya, T., Chern, J.-C.: Gesture-Based Human-Computer-Interaction Using Kinect for Windows Mouse Control and Power Point Presentation
3. Girbacia, F., Butnariu, S.: Development of A Natural User Interface for Intuitive Presentations in Educational Process. In: Conference Proceedings of eLearning and Software for Education, vol. (2) (2012)
4. Ren, Z., et al.: Robust hand gesture recognition with kinect sensor. In: Proceedings of the 19th ACM International Conference on Multimedia. ACM (2011)
5. `http://kinectpowerpoint.codeplex.com`
6. `http://www.samsung.com/sec/consumer/tv-video/tv/`
7. `http://stuffsirisaid.com/`
8. `http://www.samsung.com/global/galaxys3/svoice.html`
9. `http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx`
10. `http://en.wikipedia.org/wiki/Hooking`