# GESTURE RECOGNITION SOLUTION FOR PRESENTATION CONTROL

Darko Martinovikj    Nevena Ackovska

Faculty of Computer Science
and Engineering

Skopje, R. Macedonia

## ABSTRACT

Despite the fact that there are different presentation control techniques, today the standard mouse and keyboard are still frequently used for presentation control. Gesture-controlled solutions for presentation control also exist and they are based on motion-sensing devices like cameras, data gloves, infrared sensors and other similar devices. In this paper we present a gesture recognition solution for presentation control using the Kinect sensor. We included two gestures and studied their characteristics. For better gesture recognition we introduced 5 parameters and determined their values based on real gestures execution.

*Keywords*: presentation control techniques, kinect sensor, gesture recognition

## I. INTRODUCTION

When a speaker needs to deliver a talk or a lecture on some subject, often there is a previously prepared presentation that contains the most important notes about the subject. So, today more and more slideshow presentation software is used during a speech. To be able to control the slideshow, the speaker needs a way to input the required action. So, an appropriate presentation control technique must be chosen [1].

### *A. Presentation control techniques*

The most common and widely used presentation control is the standard keyboard and mouse input. However, this technique has some restrictions for the presenters. When the presenter needs to point some area of the slide and the projection plane is further away from the computer, then walking back and forth between the computer and the projection plane is imminent. On the other side, staying close to the computer leads to reduced body language and eye contact with the public.

Another technique that is emerging today is the usage of remote control devices and smartphones for presentation control. Because these devices have limited number of buttons, the number of different actions is also limited. This is not a big disadvantage, but as the technology progresses forward, more and more new actions starts to unveil.

Also new devices in the area of gesture recognition are getting popular and successful presentation control software has been built using this type of devices. One of them is the kinect sensor.

### *B. Kinect sensor*

Kinect sensor is an input device for motion sensing and speech recognition, developed by Microsoft [2]. This sensor allows the users to control and interact with an application using real gestures and spoken commands. With the arrival of this sensor a new way of human-computer interaction has been introduced, with a huge impact in many industries, including education, robotics, healthcare, and beyond. What made kinect so popular, compared to the other existing sensors for motion tracking, is the low price, availability to use with traditional computer hardware and existence of developers' tools for kinect application development.

## II. ANALYSIS OF EXISTING MOTION-SENSING SOLUTIONS FOR PRESENTATION CONTROL

There are many existing solutions for motion sensing. Some of them are implemented in presentation control. Gesture-controlled solutions for presentation control are usually based on motion-sensing devices like cameras, data gloves, infrared sensors and other similar devices. Some of these solutions are described in the sequel.

In [3] a system using infrared laser tracking device is presented. The presenter uses a laser pointer to make the appropriate gesture and an infrared tracking device is used for gesture recognition. This system recognises circling gestures around some object, so an easier selection in a slide can be made. Presentation control can be achieved by putting appropriate objects in each slide that will represent the appropriate action.

A presentation control solution using data gloves is presented in [4]. In [5] it is shown that a model can be trained with pre-recorded gestures and using a camera it can track and recognise gesture movements. Appropriate system called PowerGesture has been built for controlling PowerPoint™ slides using gestural commands.

Our goal was, with the usage of the kinect sensor, to create an application that will track and recognise user's gestures. We successfully created an application called ki-Prez for this purpose, and it is explained in the following section.

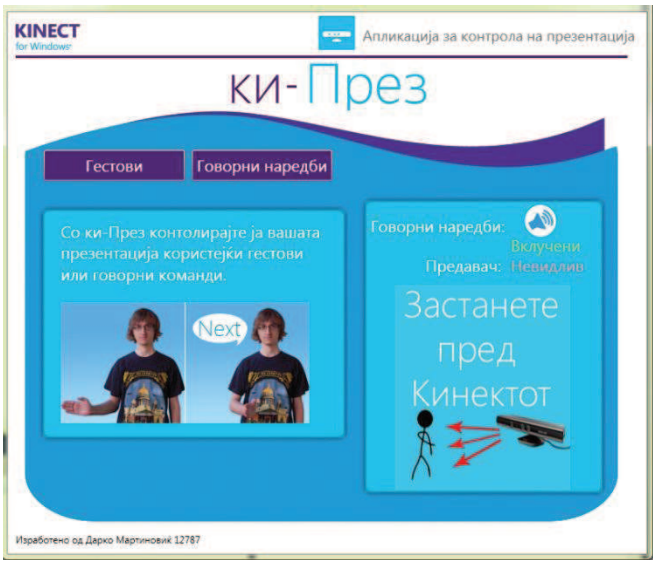
©2013 Faculty of Computer Science and Engineering

Acknowledgement: The work has been supported by the FCSE.

## III. Description of our solution

### *A. Ki-Prez as a kinect application*

We created a C# application that uses the kinect sensor for gesture control. To be able to record gesture movements, the presenter needs to stand from 0.8 to 4 meters in front of the sensor. Fig. 1 represents the graphical interface of the application.


KINECT for Windows

Апликација за контрола на презентација

ки-През

Гестови | Говорни наредби

Со ки-През контолирајте ја вашата презентација користејќи гестови или говорни команди.

Говорни наредби: Вклучени

Предавач: Невидлив

Застанете пред Кинектот

Изработено од Дарко Мартиновиќ 12787

Figure 1: Graphical User Interface of ki-Prez

When the application is started, an initialization process for the sensor is executed. Handling with unexpected events is also crucial, because the sensor may be used by another application in the same time or even turned off. Also, unexpected things can happen while the application is running, like accidently pulling off the power or connection cables. Therefore we added management with unexpected events: when an unexpected event happens, the user gets an appropriate message.

For developers working with the kinect sensor, libraries that speed up and help the development process can be found. One of them is the official Kinect Software Development Kit (Kinect SDK) from Microsoft, which we used in our application [6]. Using this SDK kit, three streams of data can be acquired: RGB, depth and skeleton data streams. The RGB data stream gives the colour information for every pixel, while the depth data gives the distance information between the pixels and the sensor. The skeletal data stream gives the positions of numerous skeletal data joints of the users that are in the tracking area of the sensor. The tracked skeleton joints of the user's body are shown in Fig. 2. By processing the depth stream data, the skeleton data stream is generated. For gesture detection, we have used the skeleton data stream. As the pixels colour isn't needed, the RGB data stream wasn't used.

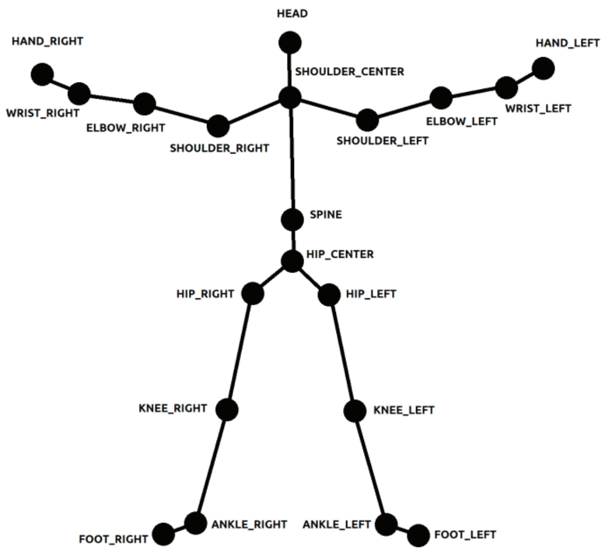


Figure 2: Tracked skeleton joints of the user's body

The stream data can be acquired in two ways: using events or polling. If the event type is used, then an application has to subscribe for the data and after that the data will be sent continuously. By contrast, the polling method gives the data on demand, when the application requests it. We have used the event-subscription type as we track the user's movements simultaneously during the presentation.

### *B. Gesture recognition*

For appropriate presentation control we have included two hand gestures: swipe left and swipe right. The practical realisation of these gestures is shown in Fig. 3.

We have analysed these two gestures and developed an algorithm for detection. The algorithm is based on the characteristics that these gestures have.

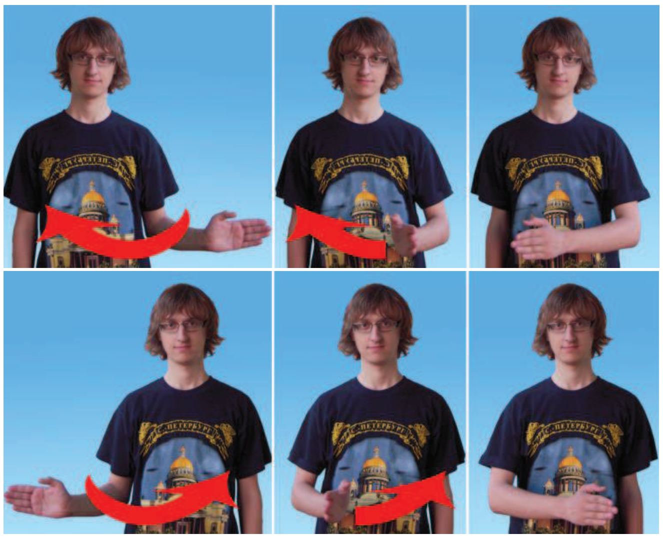


Figure 3: The two gestures: Swipe left and swipe right

©2013 Faculty of Computer Science and Engineering

Acknowledgement: The work has been supported by the FCSE.

Fig. 4 represents the coordinate system for the skeletal joint data. There are three axes, where the Z axis can have only positive values representing the distance of the joints from the sensor in meters.

Also in Fig. 4 the execution of the gesture swipe left is shown, i.e. the position of the left hand joint as the time progresses (entry's indexes are increasing).

The characteristics of the swipe left gesture that we observed are:

- The x-axis coordinate values are decreasing as the gesture is executed;
- The y-axis coordinate values have nearly equal values as the gesture is executed;
- The length of the line formed as a sum of the lengths between the points of the gesture has to exceed some previously-defined value.
- The spent time between the first and the last tracked point of the gesture has to be in the previously defined allowed range.

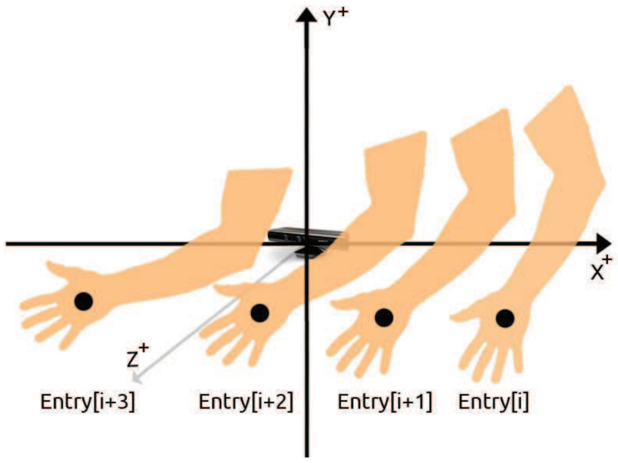


Figure 4: The skeleton coordinate system and the swipe left gesture execution

The characteristics for the swipe right gestures are the same, except for the first one where the x-axis coordinate values are increasing (not decreasing) as the gesture is executed.

*C. Parameter selection*

For every characteristic, appropriate parameter values had to be chosen, for better rate of reliability.

The parameters that we introduced are:

- $X_{max}$ maximal threshold value of the x-axis between two consecutive hand joint data expressed in meters for a recognised gesture
- $Y_{max}$ maximal threshold value of the y-axis between hand joint data expressed in meters for a recognised gesture
- $L_{min}$ minimal length of the recognised swipe gesture expressed in meters
- $T_{min}$ minimal duration of the recognised swipe gesture expressed in milliseconds
- $T_{max}$ maximal duration of the recognised swipe gesture expressed in milliseconds

For the purpose of determining the parameter values, we collected gesture data from 5 people, which executed the two gestures from different distances in front of the sensor.

Interesting conclusions were made. The parameters $Y_{max}$ and $L_{min}$ depend of the distance from which the gesture is executed. This is due to the fact that as the user is moving out from the sensor, the value deviation in two consecutive hand joint data is decreasing. For that purpose we introduced a formula for calculation of these parameters, given in (1, 2) where D is the distance of the presenter from the sensor, and $L_{min}$ and $Y_{max}$ are minimal length and maximal threshold respectfully.

$$L_{\min} = 0.8 - 0.2\frac{D}{5} \quad (1)$$

$$Y_{\max} = 0.1 - 0.05\frac{D}{5} \quad (2)$$

The duration of the executed gesture varies mainly from the gender and age of the presenter, so the parameter values of the minimal and maximal duration were chosen appropriately: $T_{min}$=350ms and $T_{max}$=1500ms.

To properly calculate the value of $X_{max}$ we considered the amount of time between two skeleton joint data generation. As the skeleton data is calculated using the depth image data from the sensor, the time for getting two successive skeleton data varies, depending on the processing power of the computer where the application is executed.

We have tested the application on Intel Core 2 Duo, 2.4 GHz, and the average time between two successive skeleton data is ≈ 40 ms. Because $T_{max}$ is 1500ms, one gesture can be consisted of maximum 1500/40=37.5≈38 successive skeleton joint data. So, the last 38 generated skeleton hand joint data are tracked, and the previous are disposed.

$X_{max}$ is calculated by 0.8 (maximal length of swipe gesture) / 38 (maximal number of successive joint data in a gesture) / 2 (time variations in getting successive skeleton data) ≈ 0.1.

To detect a swipe gesture, the successive skeleton hand joint data must be checked and when the data satisfies all of the

©2013 Faculty of Computer Science and Engineering Acknowledgement: The work has been supported by the FCSE.

parameters, gesture is detected. This is explained in details in the next two paragraphs.

For keeping the skeleton hand data two queues were used, one for the right skeleton hand joint data (left swipe gesture), and the other for the left joint data (right swipe gesture). The maximal number of elements in the queues is 38 (the maximal number of successive joint data in a gesture). When new skeleton data arrives, the hand joint data is added to the queues.

The last two skeleton joint data entries are checked for the parameters $X_{max}$ and $Y_{max}$ for both gestures. If these parameters aren't satisfied then the data from the appropriate queue is erased. If they are satisfied, then the other three $L_{min}$ $T_{min}$ and $T_{max}$ are also checked. If they are satisfied then a swipe gesture is detected.

When a gesture is detected, then an appropriate pressing of a keyboard button is simulated. The left swipe gesture represents pressing of the left arrow, and the right swipe gesture the right arrow on the keyboard.

## *D. Choosing the presenter*

According to the Kinect SDK [6], current version supports tracking of only two skeletons simultaneously. We propose the presenter to be chosen according to the minimal Euclidean distance, measured between the skeletal hip center joint and the sensor. In (3) the x, y and z represent the coordinates of the skeletal hip center joint for the appropriate skeleton.

$$\min(\sqrt{x_0^2 + y_0^2 + z_0^2}, \sqrt{x_1^2 + y_1^2 + z_1^2}) \qquad (3)$$

## *E. Speech commands*

The Kinect SDK also allows working with the sensor's audio data. So, in addition to the gestures, using the Kinect Speech Recognition Engine [6] we defined a grammar - speech commands for presentation control. In Table 1 the defined speech commands and their actions are shown. The commands simulate pressing of the appropriate button, according to PowerPoint™ slideshow software.

## IV. Evaluation

We have evaluated our gesture-recognition module measuring the rate of success of the executed gesture from different sensor distances. In Fig. 5 that rate of success is shown. It can be seen that a high success rate had been achieved from all of the distances. There is little degradation in rate of success as the presenter is getting toward the furthest allowed distance from the sensor, because the skeleton data is prone to errors, especially when the presenter is moving far away from the sensor.

Table 1: Speech commands and their actions

| Command | Action |
|---|---|
| Next | The presentation goes to next slide |
| Previous | The presentation goes to the previous slide |
| Start | The presentation is started |
| Continue | The presentation is continued from the current opened slide |
| Exit | The presentation is closed |
| White screen | The screen switch to white |
| Black screen | The screen switch to black |
| Audio on | The speech commands can again be used (if they were deactivated) |
| Audio off | The speech commands are deactivated. Only audio on can be used |
| Show ki-Prez | The interface is shown in front-end |
| Hide ki-Prez | The interface is minimised |
| Gestures info | The info block for gestures usage and information is shown |
| Speech info | The info block for speech commands usage and information is shown |

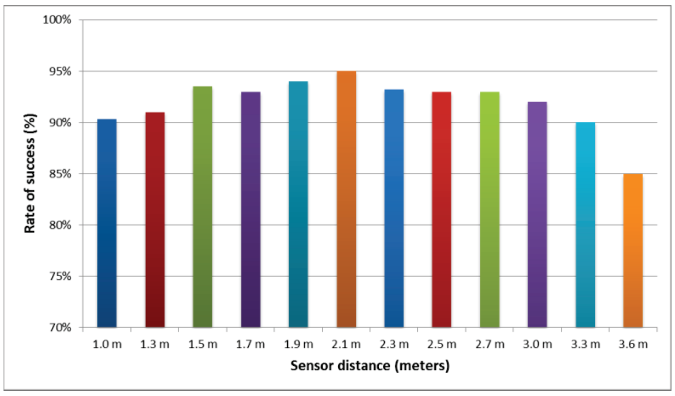


Figure 5: Results of distance evaluation of the gesture recognition

We have also evaluated the average CPU usage of the application. The testing was done on Intel Core 2 Duo, 2.4 GHz processor. These results are shown in Fig. 6 and we concluded that CPU usage mostly depends of the skeleton data. When the kinect is powered off and no skeleton data is received, then CPU usage is minimal. There is an increase in CPU usage when the kinect is plugged in, but the presenter is not in the viewing

©2013 Faculty of Computer Science and Engineering

Acknowledgement: The work has been supported by the FCSE.

range of the sensor. The CPU usage is more increased when the presenter is in front of the sensor and the application needs to process the skeleton data.

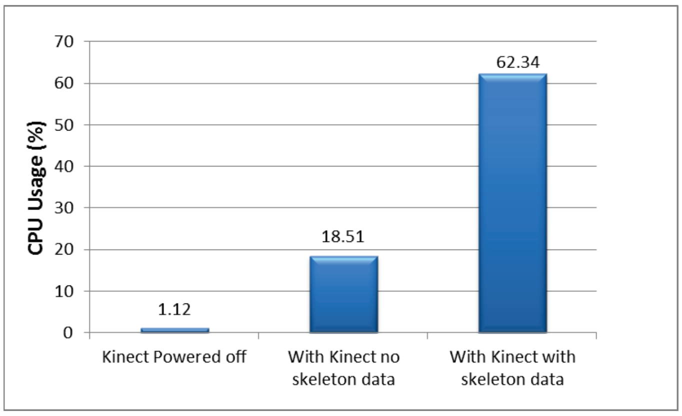


Figure 6: CPU Usage of the application

## V. CONCLUSION

In this paper we presented a gesture recognition solution for presentation control. We used Kinect sensor, and utilized it for PowerPoint presentation control. Two gestures were included and their characteristics were studied. We introduced 5 parameters for better gesture recognition and determined their values based on real gestures execution.

As a future work, we are currently studying well-known models for time-spatial data and we plan to create a gesture pattern recognition using the kinect sensor.

## REFERENCES

[1] X. Cao, E. Ofek and D. Vronay, "Evaluation of Alternative Presentation Control Techniques" *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, pp. 1248 – 1251, 2005.

[2] Kinect [Online] Available: http://www.microsoft.com/en-us/kinectforwindows/

[3] K. Cheng, K. Pulo "Direct Interaction with Large-Scale Display Systems using Infrared Laser Tracking Devices" *APVis '03 Proceedings of the Asia-Pacific symposium on Information visualisation*, vol. 24, pp. 67–74, 2003.

[4] M. Bhuiyan and R. Picking, "Gesture-controlled user interfaces, what have we done and what's next?" *Journal of Software Engineering and Applications*, vol. 4, no. 9, pp. 513–521, September 2011.

[5] H.-K. Lee and J. H. Kim, "An HMM-based threshold model approach for gesture recognition" *The IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 961–973, October 1999.

[6] J. Webb and J. Ashley "Beginning Kinect Programming with the Microsoft Kinect SDK", 2012

©2013 Faculty of Computer Science and Engineering Acknowledgement: The work has been supported by the FCSE.