Developing a Gesture Library for Working in a Virtual Environment

D. D. Tsoy, Ye. A. Daineko(✉), M. T. Ipalakova, A. M. Seitnur, and A. N. Myrzakulova

International Information Technology University, Almaty, Kazakhstan
y.daineko@iitu.edu.kz

Abstract. From year-to-year virtual reality conquers more and more IT market share and according to the forecasts, it will continue further. Except for virtual reality itself, there is also a need for the creation of new interaction ways. The specific feature of the VR requires special methods and tools that will provide users with comfort, efficiency, fast response. Also, it is important to make these tools native and understandable to answer the needs of consumers.

In the following paper, the authors present an analysis of works that suggest new types of human-computer interaction. The necessity in searching for new methods in communication between users and the virtual environment, and the need for their inclusivity and accessibility are shown.

Besides, the self-developed product and the process of a gesture library creation are reflected in the paper. It is assumed that the approach will improve the perception of information, shorten the training time for the user, which together will increase the efficiency of the application in which the product will be implemented.

Keywords: Virtual environment · Gesture · Human-computer interaction · Program application

1 Introduction

Human-computer interaction is one of the main aspects that appeared with the creation of the personal computer. Since then, people have been tending to ease the process of computer control. Nowadays we have plenty of types of different manipulators and controllers but almost all of them require specific skills to work with. From this point of view, the most interesting approach in the field of human-computer interaction is gesture control because it does not need additional effort or equipment. Also, this approach is available for a broad group of people and can be used in different areas. The reason is gestures are simple and natural, even for people with some impairments.

So, one of the most popular devices of this type is Leap Motion Controller (LMC). It tracks the movement of hands without any physical contact. There are two infrared cameras and three LEDs within Leap Motion which collect the data that is then processed by software.

Many researchers use the controller in their studies that demonstrates the versatility of the device. For example, [1] provides a study and its results in the development and

© Springer Nature Switzerland AG 2021
L. T. De Paolis et al. (Eds.): AVR 2021, LNCS 12980, pp. 17–24, 2021.
https://doi.org/10.1007/978-3-030-87595-4_2

definition of gestures based on American Sign Language. The authors pointed out the lack of such studies, justified using a single sensor, which leads to low-quality data and, as a result, to an incorrectly trained neural network. For this reason, the researchers used a multi-sensor system. Then, by overlaying the extracted data, a set was obtained for training a neural network based on a hidden Markov model. The results of the work demonstrate the effectiveness of this approach compared to existing systems.

In [2], the authors also used a neural network to process data obtained with the Leap Motion controller. Since the study is based on the determination of a combination of gestures, the SVM algorithm was used. The full process consisted of three parts. In the end, the authors obtained 63.57% of the accuracy of gesture recognition in real-time, and the system determined gestures shown not only by hands but also by fingers.

Another trend in the natural user interface world is the use of the Leap Motion controller as a means of authorization and authentication. So, according to the [3], the controller can be used equally with traditional means of information input: a mouse and a keyboard. The paper also provides information on the uniqueness of the behavior of each user during the execution of a given gesture. The results of the work demonstrated high accuracy rates, which confirmed the authors' assumption about the effectiveness and high potential of using the Leap Motion controller.

The following work [4] describes a study on the use of Leap Motion as a video player control device. The authors have created a framework for blind and visually impaired people, based on the definition of gestures for performing video operations. Their results demonstrate the effectiveness of this approach to interacting with software for people with vision problems. Thus, according to the respondents of the study, control with the help of gestures in 35% of cases was assessed as "very easy" and "easy" in 65%.

Another interesting example of implementing gesture recognition to interact with a program is the Scratch extension [5]. It allows the user to implement the controller's functionality into his project and, for example, control the movement of the character through the movement of his fingers. Figure 1 shows an example of the interface of this program. However, the extension only allows you to use standard Leap Motion gestures.

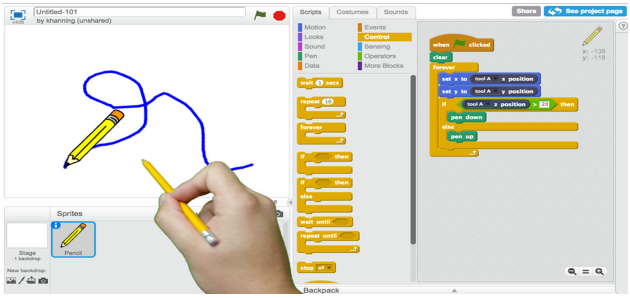

Fig. 1. Scratch app interface.

There is also an extension that allows managing the elements of the Google Chrome browser using Leap Motion [6]. A user, after the controller is connected, can scroll pages (vertically and horizontally), refresh pages, zoom in/out, navigate through history, and

switch tabs. Also, it is possible to configure the controller parameters for personalization, when the user can change the set of gestures, notifications mode and sensitivity coefficient.

The authors in [7] demonstrated results of the project on dynamic gesture recognition approach in medicine. To define gestures, the SVM algorithm also was used. In result authors got the library with eleven dynamic gestures and overall performance of the model 80%. Within the work they described each of the gesture definition step. In general, the paper shows how LMC can ease the work of the stuff in areas where contactless control is required.

The following work [8] describes gesture elicitation study during which participants used some gestures and evaluated their difficulties for different tasks of TV control. The authors have given a list of commands, tasks, and results of research where different features of gesture performance were tracked. This project allows better understanding of features that are important in gesture recognition and execution.

2 Prerequisites for Development

The authors' work on virtual physical laboratories raised the problem of user interaction with objects within the virtual environment. One of the main tasks during the development of these laboratories is to choose the way of material presentation. It is important because users should get certain experience and knowledge about experiments after the execution of a laboratory work. In the case of a virtual laboratory, the quality of interaction between application and the user depends on the naturalness of the interface. Thus, an experiment conducted with one's hands improves the perception of the material being studied and makes the experience of working with the program brighter and more memorable.

Based on these features the Leap Motion controller was chosen as an alternative to a mouse and keyboard and other types of joysticks. The virtual physics laboratory works were created within the Unity game engine. Thus, Leap Motion integration was done easily and quickly because of the special module.

However, further work with the controller revealed several drawbacks. Although operations with virtual objects are performed without intermediaries, that is, directly with the help of hands, they are not so easy to implement due to various conditions: insufficient lighting, its excess, lack of feedback from the controller, especially for people who had no previous experience working with Leap Motion.

This prompted the authors to create a library that expands the set of standard Leap Motion gestures, and, thereby, speeds up the work inside the virtual laboratory for performing experiments and facilitates the interaction between the user and virtual elements. The aforementioned drawbacks have been mitigated by the recognition of the implemented gestures. In addition, the new introduction mode within the virtual laboratory have been introduced, in which a user can learn these new gestures and receive the feedback from the system as screen messages with the information about a device, the hand that is tracked and the recognized gesture.

3 Project Description

The Leap Motion Controller is a new consumer-grade sensor. It is primarily designed to detect hand gestures and finger positions in interactive software applications.

According to its API [9], the controller recognizes four gestures (Fig. 2):

– TYPE_SWIPE – quickly move fingertip across the screen to the right and left;
– TYPE_CIRCLE – fast circular motion of fingertip across the screen;
– TYPE_SCREEN_TAP – quickly push or tap the screen;
– TYPE_KEY_TAP – quickly move fingertip from down to up on the screen surface.

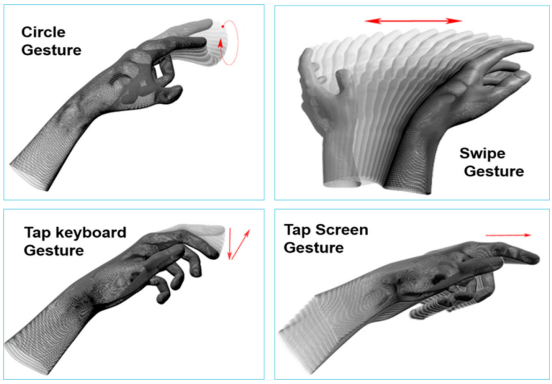

Fig. 2. API's gestures graphical representation.

Initially, it was planned to create an additional set of gestures that would allow controlling the virtual laboratory work. However, it was decided to create a library that could be used in other applications with ability to change the gestures set and redefine their functionality. Now, the authors have implemented the following gestures:

– TYPE_FLIP – flip the palm with outstretched fingers;
– TYPE_BLOOM – transition from a palm with fingers, the tips of which are gathered together, to a palm with spread fingers;
– TYPE_FIST – fist, all fingers of the hand are clenched;
– TYPE_PINCH – the tips of the index and thumb are brought together.

A graphic demonstration of the implemented gestures is shown in Fig. 3 a, b, c, and d. The gestures were selected based on their prevalence among existing analogs, as well as considering the ease of execution and ease of memorization.

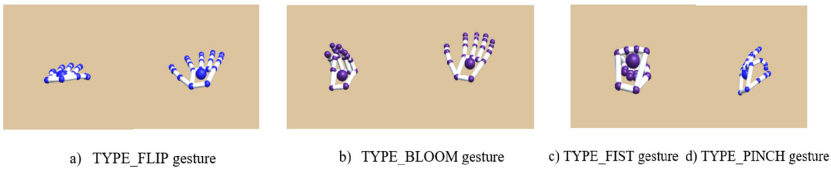

Fig. 3. Graphical representation of new gestures.

4 Project Structure

The library was developed in C#, Fig. 4 shows an activity diagram that demonstrates how the library can be used in an application. The user launches an application using a library. Then he or she can override given gestures or to customize them to suit user's needs.

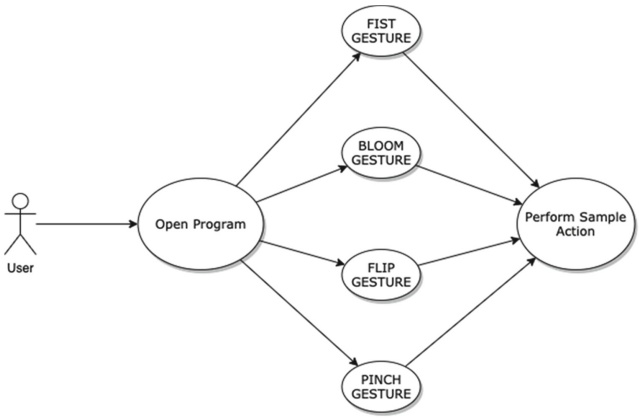

Fig. 4. Activity diagram.

Figure 5 shows the relationship between the classes that were implemented in the library. There are four classes on the diagram: PinchGesture, FlipGesture, FistGesture, and BloomGesture, which are based on the FrameworkGestures class. FrameworkGestures class defines their basic parameters. Thus, they are in a generalizing relationship. Then, each class describes its conditions to track a particular gesture. Finger Class, Arm Class, Hand Class are in an association relationship.

The classification of a human hand bones corresponds to the classification in Leap Motion. Bones are identified as:

- metacarpal bone – the bone within the hand that connects the finger to the wrist (excluding the thumb).
- proximal phalanx – the bone at the base of the finger, connected to the palm.
- intermediate phalanx – the middle bone of the finger between the tip and the base.
- distal phalanx – the terminal bone at the end of the toe.

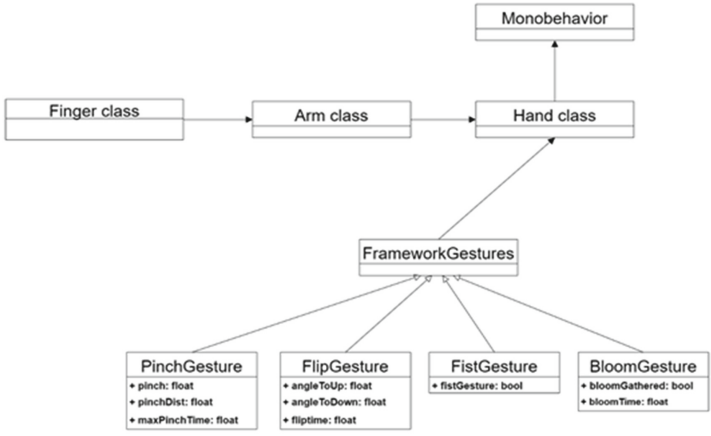

Fig. 5. Class diagram.

Depending on the gesture and the bones required to define it, the angles between the bones and the duration of the gestures are calculated.

5 Pinch Gesture Implementation

To define the gesture, the Hand class and its pinchStrength attribute are used, which returns a float value from 0 to 1 and indicates the pressure between the thumb and one of the four remaining fingers. Since in the library the Pinch gesture is defined as squeezing the index and the thumb, in the function we calculate the distance between them to make sure that the definition is correct.

6 Bloom Gesture Implementation

Bloom gesture tracking is done by calculating the distance between the knuckles. Having set a value for the maximum allowable distance, we track its change and, if it is done, we check the implementation of the palm gesture. Thus, the gesture consists of two sequential actions: bringing the fingertips together and then opening the palm. If the second stage is not performed, then the gesture is not considered as executed. Therefore, the duration of these movements one after another in a certain period is also important.

7 Fist Gesture Implementation

Leap Motion has a built-in boolean function isExtended, which determines whether the finger is extended. If it is, the return value is true. The finger will not be extended when it is bent towards the palm. The palm containing the instruments will always return a positive value. We count the fingers that are not extended, and if all fingers are bent, the gesture is regarded as Fist.

8 Flip Gesture Implementation

To implement this function, the Vector3 structure was used, which allows to operate with vectors and points in three-dimensional space. This structure is used to convey 3D positions and directions. It also contains functions for performing general vector operations and has many pre-defined properties for specifying direction, tilt, and rotation, and transforming vector points. In this example, the angle of rotation of the palm is calculated at a certain time, which allows tracking the rotation of the palm with outstretched fingers. As with the Bloom gesture, the sequence of movements is important in the current one. Therefore, we first track the palm with the fingers outstretched, turned back down, and then turned up. If the actions are performed sequentially, then the gesture is considered as completed.

Thus, now there is an implementation of the definition of four gestures that you can override and perform the actions necessary to work in the virtual world. It is planned to expand the library and create an interface, as well as write documentation to involve stakeholders and improve the project.

Now, the framework has been implemented in the virtual laboratory work on the topic "Investigation of the flight range of the body from the throwing angle". Its use helped to simplify the placement of the ball in the gun tray for throwing, as well as the user interface interaction required to launch the gun and reload the application to invoke reference materials. Thanks to the ability to override gesture functions, the framework becomes more flexible and easily customizable for an individual user. To assess and determine the effectiveness, it is planned to compare the results of users using the application without and with the framework.

9 Conclusion

Every year, more and more new projects appear that help various groups of users to simplify the work with the software, making it native and accessible. This article describes the latest developments in gesture control. The work presents the gesture library developed by the authors, which is an extension of the existing functionality of the Leap Motion controller.

Further work includes expanding the list of possible functions based on raw data from Leap Motion sensors, as well as analyzing two-handed gestures and dynamic gestures. This will greatly increase the intuitiveness of the system. Also, this functionality is planned to be implemented in augmented reality learning systems to automatically detect gestures and the user's actions. In the course of work on the project, the need for a survey to identify the most frequent and simple gestures was identified. This will increase the stability of the system and supplement it with many available movements.

Acknowledgment. This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan (Grant No. AP08857146).

References

1. Fok, K.Y., Ganganath, N., Cheng, C.T., Chi, K.T.: A real-time ASL recognition system using leap motion sensors. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 411–414. IEEE (2015)
2. Kumar, P., Saini, R., Behera, S.K., Dogra, D.P., Roy, P.P.: Real-time recognition of sign language gestures and air-writing using leap motion. In: Fifteenth IAPR International Conference on Machine Vision Applications (MVA), pp. 157–160. IEEE (2017)
3. Chan, A., Halevi, T., Memon, N.: Leap motion controller for authentication via hand geometry and gestures. In: Tryfonas, T., Askoxylakis, I. (eds.) HAS 2015. LNCS, vol. 9190, pp. 13–22. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20376-8_2
4. Funes, M.M., Trojahn, T.H., Fortes, R.P.M., Goularte, R.: Gesture4All: a framework for 3D gestural interaction to improve accessibility of Web videos. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, pp. 2151–2158 (2018)
5. Scratch Leap Motion Extension. https://khanning.github.io/scratch-leapmotion-extension/. Accessed 18 Feb 2021
6. Leap of Faith Chrome Extension. https://github.com/Issam-b/leap-faith. Accessed 21 Feb 2021
7. Ameur, S., Khalifa, A.B., Bouhlel, M.S.: A comprehensive leap motion database for hand gesture recognition. In: 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pp. 514–519. IEEE (2016)
8. Zaiţi, I.A., Pentiuc, Ş.G., Vatavu, R.D.: On free-hand TV control: experimental results on user-elicited gestures with leap motion. Pers. Ubiquitous Comput. **19**, 821–838 (2015)
9. Leap Gesture. https://developer-archive.leapmotion.com/documentation/python/api/Leap.Gesture.html. Accessed 14 Mar 2021