Solving 3D Virtual Rubik's Cube with Gestures

Basheeruddin Baig
Basheer5@colostate.edu
Group Name: 3D Virtual Rubik's Cube

Abstract

In this project the presentation is about solving the Rubik's Cube with Interactive methods.The common interactive inputs for these applications are either a mouse or a keyboard but here in this paper hand gestures are used to manipulate the objects. The idea of this project is to replicate the functionality of the mouse with your own hand and doing all the tasks that a mouse can perform. The game is used as a tool to compare the performances between the mouse and hand gestures. It is a simple but novel idea as many users who have muscular disability can cannot hold objects like mouse. This would be a perfect solution to them to interact with their system using nothing but their hands. The only requirement is to have a camera integrated to the system which detects the hand gestures. Rubik's cube is very interactive game and using tools like mouse or keyboard dilute the experience. In the experiment users can solve the 3D Rubik's cube with free hand moments. This is a simple concept with correct implementation you will no longer need a mouse in a system. It is similar to a touch screen device but here you need not even touch it.In this project we will develop a virtual 3D Rubik's Cube using Unity game development platform and solving that Cube with using mouse and hand gestures. I have used Machine Learning models to train and recognize my gestures and a wireless laser mouse WM126.

1 Introduction

Human interactive input device has become an essential study issue as computer technology have advanced and the number of computing applications has grown. For very diverse gameplay requirements, additional sorts of input devices are required in computer game creation [7]. Rather than relying on traditional input devices such as the mouse, keyboard, joystick, and more advanced devices such as touch screens, the human body could potentially serve as a "input device" directly in the form of human interactive input interface, which is becoming increasingly popular in the video game industry. One enticing aspect of human interactive input interface is that it allows people to connect with computers without the usage of physical control devices such as a keyboard and mouse.

Rubik's Cube games are generally accessible on PCs and mobile devices. The engagement in these platforms is not as intuitive as it is in reality, which is playing the Rubik's Cube using a mouse, a touch screen, or keyboard inputs. [1] To use the human body as an input device, human motion must be recorded, which might range from finger motions to arm movements. The majority of present methods for capturing human body motion include the attachment of sensors to the human body, such as the Nintendo Wii game system or a data glove. While the connection of sensors allows for the quick capture of the item's movements, it also acts as a control interface barrier between the actual object and the underlying game program [8].

Alternative to devices and sensors is computer able to recognize human location and movements. A vision-based technique to provide a distant, non-contact mouse control interface is presented in this work. [9] In the framework of the WIMP paradigm, this perceptive interface integrates common web-cameras with powerful vision methods and allows the user's hands to subsume the hardware pointing devices. [3] Gesture Controlled Virtual Mouse simplifies human computer interface by utilizing Hand Gestures. There is almost no direct touch is required with the computer. Virtually all input-output operations may be controlled using static and dynamic hand gestures. To identify hand motions, this study employs cutting-edge Machine Learning and Computer Vision techniques. This runs well without the need for any extra hardware. It makes use of models such as CNN, which is implemented by MediaPipe and runs on top of pybind11. The application of hand gesture works directly on hands by making use of MediaPipe Hand detection and you can also use any glove and operate it.

2 Related Work

Alternative input devices were often used in the 1990s to refer to headmount devices and data gloves. A head-mount device is a hefty device worn over the head that displays visuals to the user and is capable of monitoring head position and ocular movement [4]. This might be thought of as an alternative to the mouse, in which ocular movement is converted into mouse movement. The data glove, which has multiple sensors mounted to the back of the hand, is another common device for recording gesture-based input. It makes it simple to determine the exact location and orientation of the hand at any time by measuring the joint angles of each finger and thumb, which are then utilized to create the gesture for computer input [4].

The computational capability of computers has surpassed real-time body tracking and recognition through video stream due to fast developments in powerful processors. In this project, we look at how hand tracking and gesture recognition may be used to not only emulate pointer devices like the mouse, but also give gesture instructions that can be used by programs. Gesture recognition [10] [2] is an intriguing form of real-time monitoring application that uses mathematical algorithms to comprehend human motions. Body language and sign language are interpreted by hand gesture recognition algorithms.

Image filtering, hand tracking, and gesture identification are the three primary phases in recognizing hand movements from video feeds. [5] Hand tracking determines and limits the location of the hand within the frame. Gesture recognition quantifies a hand picture that has been tagged with hand position and boundary information in order to detect the intended gesture. Picture filtering for hand recognition is in charge of removing image segments that are not part of the hand motion.

Leap Motion is a revolutionary concept for playing the Rubik's Cube with hand motions. Furthermore, it allows a user to explore new and unique ways of playing Rubik's Cube while maintaining the reality of playing Rubik's Cube.[2]

The Oculus Rift software with the Xbox Controller: This program for solving the Cube makes use of the Oculus Rift as an output device and the XBox controller as an input device. [6] The application's goal is to provide a novel approach to solve it in the virtual environment by utilizing instructions that signal and create movements of the Cube as a whole, as well as movements of the selected side.

3 Literature Survey

3.1 Rubik's Cube

Rubik's Cube is a cube-shaped puzzle game. It is available in six various colors: white, blue, green, red, orange, and yellow. One color is shown on each surface of the cube. The conventional dimension for the Rubik's Cube is 3x3x3, however several additional dimensions or variations have been made to allow the cube to be played in a variety of experience and complexity levels. The conventional Rubik's cube is the 3x3 Rubik's cube, which was Erno Rubik's initial version. The main idea behind Rubik's Cube is to rotate the surfaces of the cube in order to place all cubes of the same color on the same surface.

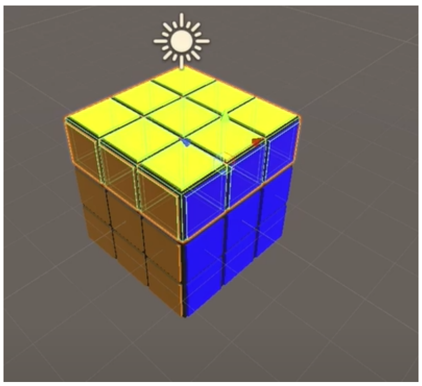

Figure 1: Rubik's Cube

3.2 Unity Game Engine

Unity allows users to build games and experiences in both 2D and 3D, and the engine provides a core scripting API in C using Mono, for both the Unity editor and games themselves, as well as drag and drop capability. Unity is very feasible in connecting with interactive objects like AR and VR modules. We need to download the respective packages and plugin the connections. There are numerous number of games that can be played on Unity using these plugins

3.3 Gesture Recognition using Camera

A gesture recognition system begins by pointing a camera at a specified three-dimensional zone within the vehicle and gathering frame-by-frame photos of hand placements and actions. This camera is often positioned in the roof module or another unobstructed observation point. Even when there isn't much natural light, the system lights the region using infrared LEDs or lasers to provide a crisp image. [9] Computer vision and machine learning algorithms analyze the photos in real time, converting hand gestures into instructions based on a predefined library of signals.

4 METHODOLOGY

The approach that is used to design the project application as a guideline. There are three parts which are Rubik's cube game, Gesture recognition part and the experiment part to derive the conclusions.

4.1 Rubik's Cube Game

Unity game engine was used to build the game from scratch. 3D cube object was created and each cube was given a face, side and bottom. Cube can be rotated using mouse actions for the user to look around the cube and analyze it. Player can interact with the cube with the ability to rotate any side of the cube using mouse. There is option to select shuffle and solve button if the user wishes to use.

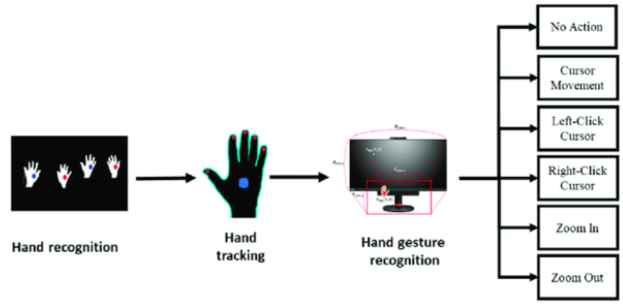

Figure 2: Hand Gesture Workflow

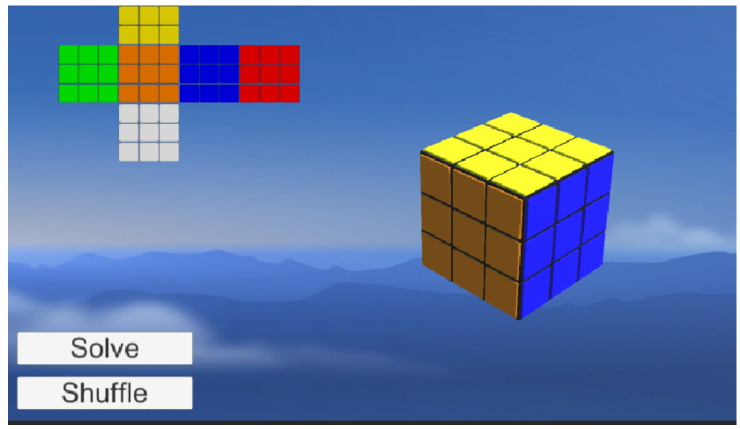

Figure 3: Game UI

4.2 Gesture Recognition using Camera

User can control the cursor by using hand either left or right. Open hand palm is recognized and neutral gesture which means it does not do any action. Gesture "V" is used to control the mouse, In this gesture only index and middle finger is open and closing all other fingers. The gesture is also dynamic which means faster the movement of the hand faster the cursor moves.

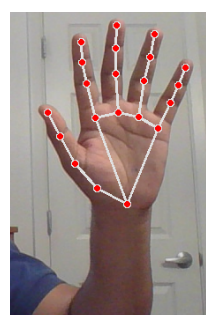

Figure 4: Neutral Gesture

Right click operation can be executed by keeping the hand in "V" gesture and blinking the middle finger once. Left click operation can be executed similar to right click but difference is we should blink the index finger.

Drag and drop action can be performed by first keeping the hand in neutral gesture which is palm open and then closing it into fist and moving the closed fist to desired location

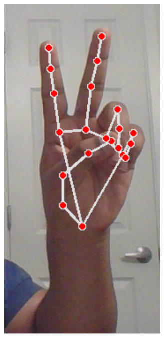

Figure 5: Cursor moment gesture

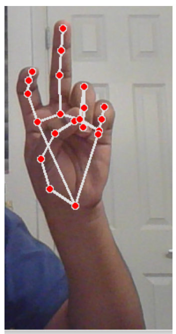

Figure 6: Left click

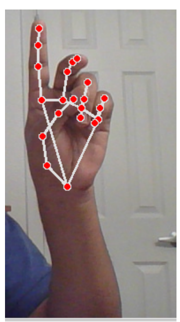

Figure 7: Right click

Figure 8: Drag and drop

The above gestures are mainly used to play the game but there are other gestures which you can see in the below figures like volume control and Multiple item selection.

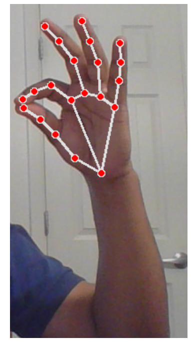

Figure 9: Volume Control

4.3 Experiment

Experiment is carried out to compare between mouse and virtual gestures (Virtual mouse). The experiment is to play the rubik's cube game. There are three levels in the game based on number of moves required to solve the game. Easy level requires two to three moves to solve the cube. Medium level requires four to five moves to solve and Hard level requires around six to seven moves.

Time is taken as a metric, Time required to complete each level is noted down to perform ANOVA analysis. After completion of the game users are asked to perform all the gestures once and then they are required to fill a survey which has series of questions and based on the feedback arrived opinion analysis was performed.

Total number of test subjects are twelve and they are from different backgrounds and age. There are four participants aged under 15, four participants aged between 15-30 years old and four participants above age 30. This distribution among the test users was taken to check if age is a factor in playing the game and performing the gesture tasks.

4.3.1 Hypothesis

Below are the hypothesis taken and conclusions will be derived after statistical analysis is performed on the data received from the users.

In the experiment conducted with mouse and gestures recognition application. Three levels of games are used which are easy, medium and hard. A two-way ANOVA test will be performed to test the below three null hypotheses.

- H1: There is no significance difference of means between mouse and gesture method
- H2: There is no difference between the means of easy, medium, hard levels of the game.
- H3 : The level of difficulty does not interact or intervene with mouse or gesture method.

5 Data Analysis

Below data given in the tables (Table 1 and 2)is collected from the users. Those values are in seconds. Each participant plays the game in three different levels and the time is recorded.

5.1 Anova Analysis

To analyse and verify our hypothesis ANOVA two-way method was applied on the users data and below you can see the ANOVA table(Table 3)

Table 1: Experiment data from Test Subjects using mouse

Participant no	GameLevel-easy	GameLevel-Medium	GameLevel-Hard Values in seconds
1	10	25	44
2	14	19	54
3	15	26	39
4	16	24	50
5	10	26	38
6	8	30	54
7	12	33	43
8	12	34	38
9	15	28	52
10	13	29	49
11	14	27	48
12	12	28	42

Table 2: Experiment data from Test Subjects using Gesture Application

Participant no	Easy	Medium	Hard
1	12	24	45
2	10	29	40
3	14	25	44
4	12	27	40
5	13	21	38
6	15	24	41
7	12	30	45
8	10	25	50
9	16	30	43
10	14	29	47
11	9	19	42
12	8	20	44

5.1.1 H1 hypothesis

The first row in the ANOVA table is the Sample row from which we take F statistic and F critical values to see if there is any difference between the means, When played with mouse and gesture application. F critical is 3.989 and F statistical is 3.7843. F critical is greater than F statistical so we do not reject the null hypothesis H1. H1 hypothesis we took was that there is no difference of means between mouse and gesture method.

From this conclusion can be drawn that participants perform the task with same speed and accuracy with gestures. Although gestures is not used by them daily and they are not experienced to it like mouse. This adds more leverage to gesture method and with enough time spent on it they can outperform tasks compared to mouse.

Another way to reject H1 hypothesis is that you can see P-value is 0.0559 which is greater than alpha vale 0.05 and hence rejecting H1 hypothesis.

5.1.2 H2 Hypothesis

If we see the columns row from the table F statistical is 417.5195 and F critical is 3.135. F statistical value is larger compared to critical. So we can reject our H2 hypothesis which is there is no difference between the columns. The alternative to H2 is true which means that there is impact on the game performance (Time) when the game levels are changed from easy to hard.

This conclusion was obvious as there would be a significant change in time to complete the task as the level of difficulty increases.

Table 3: ANOVA table

Source of Variation	SS	df	MS	F	P-Value	F critical
Sample	56.888	1	56.888	3.7843	0.0559	3.989
Columns	12553	2	6276.5	417.5195	3.45E-38	3.135
Interaction	15.44	2	7.722	4.5136	0.03006	3.135
Within	992.166	66	15.032			
Total	1367.5	71				

5.1.3 H3 Hypothesis

Looking at interaction row form the ANOVA table, F statistic is 4.5136 and F critical is 3.135. F statistical is larger than F critical which concludes that H3 hypothesis is false. There is affect on the interaction method with change in the difficulty level or you can say that level of difficulty increases there performance change is observed when compared to mouse and gesture method.

It is observed from the summary tables that average time taken to complete the game by mouse is greater than average time taken by gestures this concludes that gestures are easy to perform and play the game than mouse control.

Table 4: Summary data of mouse

Summary-mouse	Easy	Medium	Hard	Total
Count	12	12	12	36
Sum	151	329	551	1031
Average	12.58333	27.416667	45.91666	28.63888889
Variance	5.71967	16.08333	36.26512	209.4944

Table 5: Summary data of Gesture Method

Summary-Gesture Method	Easy	Medium	Hard	Total
Count	12	12	12	36
Sum	145	303	519	967
Average	12.08333	25.25	43.25	26.8611
Variance	6.083333	14.93181	11.113636	177.9515

5.2 Opinion Survey Analysis

A series of questions were asked to all the 12 participants after playing the game. There were three age groups, age less than 15, age between 15-30 and age 30 and above. This was taken to see what different age group felt with interaction using gestures. Based on the survey response below are the results.

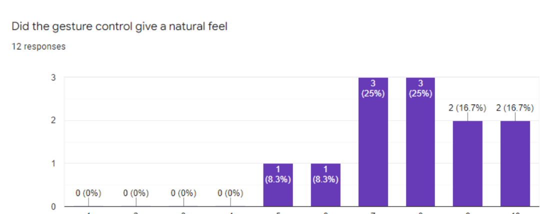

Figure 10: Participants who felt gesture was natural

Out of 12 participants 10 participants felt that gesture control gave more natural feel than mouse control

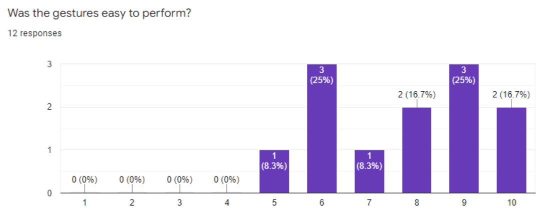

Figure 11: Easy to operate

Majority of the participants have voted that playing the game with gesture control and performing other tasks like right click and left click was easy with gestures.[Figure 11]

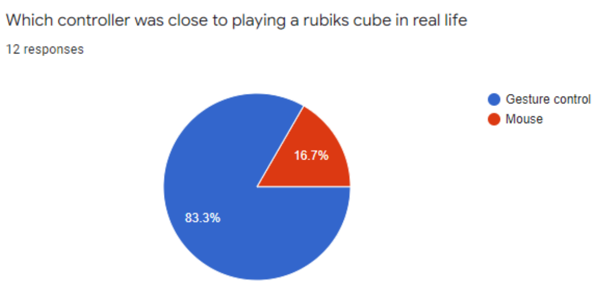

Figure 12: Controller close to real-life experience

83.3 percent of the participants voted that gesture controller was close to playing the Rubik's cube in real time. Rest 16.7 percent participants who were mostly aged above 30 years old voted that it did not give them a real life experience. I think this is because as the gaming experience they had is less compared to other age groups.[Figure 12]

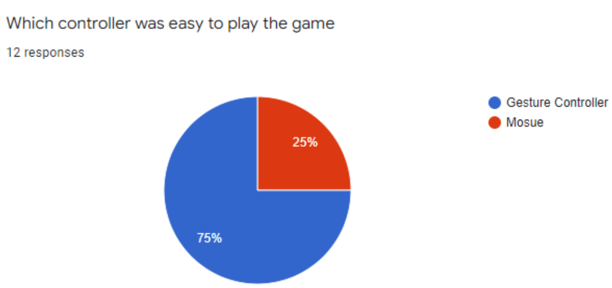

Figure 13: Ease of controlling

75 percent of the participants voted that gesture control was easy compared to mouse. As it was fast and required less movement. Other 25 percent choose mouse over gesture because they were not comfortable to keep their hand floating for a long time.[Figure 13]

When the participant performed the task to rotate the cube with both mouse and gesture the cube has to move in the direction they intend if it does not then the is a miss match in the action performed and action executed. On a scale of 1 to 10 review was taken that which performed better in terms of accuracy, here accuracy is

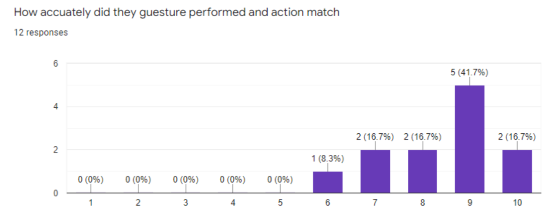

Figure 14: Accuracy feedback

achieved when the action and response are the same as the participant intended. 9 participants rated above 8 which is Majority of the participants had accuracy with gestures.[Figure 14]

As the project idea is to replace the mouse and making gestures as a key controller, Qestion was asked if they think gestures can be an alternative to replace the mouse and 9 out of 12 participants said that it could which is 75 percent of total count.[Figure 15]

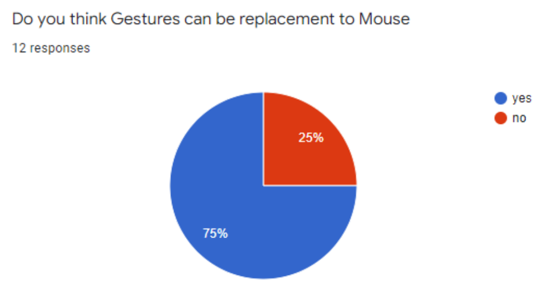

Figure 15: Participants who prefer gestures over mouse

Apart from palying the game participents were asked to do simple tasks using gestures like Right/left click, Drag and drop, Volume control, Scrolling and Double Click. Out of all these tasks drag and drop was the toughest to perform as mouse does it with ease compared with gestures.[Figure 16]

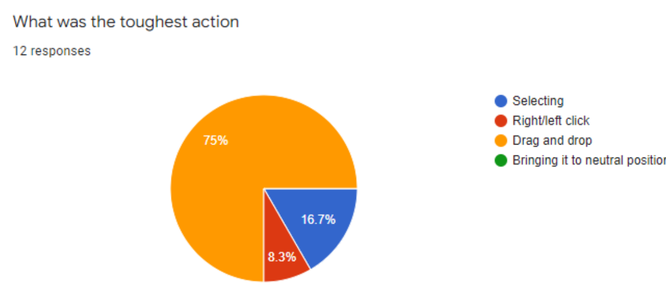

Figure 16: Toughest action to perform

6 Conclusion

This paper aims to replace the use case of a mouse with gesture control not only in playing simple games like Rubik's cube but also to perform basic operations that a mouse does. Below are the conclusions from the experiment.

From ANOVA analysis H1 hypothesis was true that is there is no significant difference between the task performance when gesture or mouse was used.

H2 hypothesis was false which concludes that there is affect in performance of the task when the level of the game is increased in both gestures and mouse.

H3 hypothesis is also false from our analysis and concludes that as the level of difficulty increased gesture control method was performing better than mouse controller.

Based on Opinion survey we can say that majority of the participants are comfortable with using gesture method. 83.3 percent of the participants felt that gestures gave them a natural feel than playing with mouse. Ease of operation was good with both the methods but mouse was more comfortable with participents that were aged above 30.

7 Improvements

Task like drag and drop and multi select needs to be improved with better gestures and faster action. This was the feedback based on survey questions. Gestures should not be complex and every age group should feel comfortable in using them. Mouse has a advantage that our hand does not strain while working for a long period of time. Gestures to some extend does the task fast and accurately but cannot be preferred for a longer time.

Acknowledgments

I would like to thank my professor Dr.Francisco R. Ortega for guiding and helping me through out the course. Also my Teaching Assistances Jachinda Li and Richi Rodriguez for assisting me through out the course.

References

[1] K. S. Abhishek, L. C. F. Qubeley, and D. Ho. Glove-based hand gesture recognition sign language translator using capacitive touch sensor. In *2016 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp. 334–337. IEEE, 2016.

[2] A. A. Argyros and M. I. Lourakis. Vision-based interpretation of hand gestures for remote control of a computer mouse. In *European conference on computer vision*, pp. 40–51. Springer, 2006.

[3] A. T. Chan, H. V. Leong, and S. H. Kong. Real-time tracking of hand gestures for interactive game design. In *2009 IEEE International Symposium on Industrial Electronics*, pp. 98–103. IEEE, 2009.

[4] A. T. Chan, H. V. Leong, and S. H. Kong. Real-time tracking of hand gestures for interactive game design. In *2009 IEEE International Symposium on Industrial Electronics*, pp. 98–103. IEEE, 2009.

[5] H. Cheng, L. Yang, and Z. Liu. Survey on 3d hand gesture recognition. *IEEE transactions on circuits and systems for video technology*, 26(9):1659–1673, 2015.

[6] Y. Fang, J. Cheng, K. Wang, and H. Lu. Hand gesture recognition using fast multi-scale analysis. In *Fourth international conference on image and graphics (ICIG 2007)*, pp. 694–698. IEEE, 2007.

[7] T. A. Garan and N. M. Suaib. Hand gesture integration of 3d virtual rubik's cube using leap motion.

[8] R. C. Luo and M. Chiou. Hierarchical semantic mapping using convolutional neural networks for intelligent service robotics. *IEEE Access*, 6:61287–61294, 2018.

[9] G. Murthy and R. Jadon. A review of vision based hand gestures recognition. *International Journal of Information Technology and Knowledge Management*, 2(2):405–410, 2009.

[10] A. Van Dam. Post-wimp user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.