Gesture Recognition in HCI

Karthikeshwar
Student, BTech in Computer Science & Engineering
REVA University
Bengaluru, India
karthikeshwar7087@gmail.com

Siddhanta Mandal
Student, BTech in Computer Science & Engineering
REVA University
Bengaluru, India
siddhantamandal2015@gmail.com

Salony Shah
Student, BTech in Computer Science & Engineering
REVA University
Bengaluru, India
r18cs529@cit.reva.edu.in

Prof. Nikhil S Tengli
Assistant professor, School of Computer Science & Engineering
REVA University
Bengaluru ,India
nikhil.tengli@reva.edu.in

Abstract— **Our work is about gesture recognition software and its applications in Human-Computer Interaction (HCI). We aim to develop a software which can enable the user to control the computer with just hand gestures. This makes interaction with the computer more intuitive and natural. Computers are being used more than ever before and the complexity of user applications and use cases has also increased. This demands for new ways of HCI. Augmented and Virtual Reality (AR & VR) is also being touted as the future of computing, which means the traditional methods of screen touching, mouse and keyboards can't be used in those cases. In that case, hand gestures can be very helpful. Our gesture recognition software shall be a GUI application, developed for Windows, macOS and Linux (major distributions). The app can be used to customize what action a gesture performs. The user can also add a custom gesture and assign a specific task to that gesture.**

Keywords-computer vision; gesture recognition; human-computer interaction

I. INTRODUCTION

Gesture Recognition technology has been shown in movies (like Minority Report) as a sci-fi technology of the future [1]. But it failed to gain traction due to various reasons: (a) Lack of technology, powerful enough hardware [2]. (b). Even with good hardware, recognizing and tracking the gestures is hard, owing to (i) background noise, (ii) poor lighting conditions, (iii) different gestures recognition [3]. Social acceptability is also one of the main issues [4]. This is why popular products like Microsoft Kinect [5] and Leap motion controller failed [6].

But the main interaction methods of the personal computer has remained the same since Apple launched the Apple II in 1977. But the usage of computers and the use cases has increased in complexity [7]. Also, according to various research areas and R&D news leaks, Augmented and Virtual Reality (AR & VR) are said to be the future of computing [8]. When it comes to AR & VR, traditional methods of HCI like the mouse, keyboard and even screen touch technology is of much lesser help. Hence there is a need for new ways of HCI and the most natural one among them is hand gesture recognition.

II. LITERATURE SURVEY

A. Types of Gesture Recognition

There are various ways in which gesture recognition can be done: (A) Hand gesture recognition (B) Facial gesture recognition (C) Sign language recognition (D) Device gesture technology (recognizing gesture with the help of a device attached to the user, e.g. a wrist band) (E) Electric field sensing [3]. Mainly, two types of gesture recognition are possible: (i) Offline gestures: gestures are processed after the user has interacted with the device. (ii) Online gestures: Gestures are processed on-the-go, used mainly for directly manipulating objects/cursor/pointer/etc. [9] The

whole idea of gesture recognition stands is based on the idea of Touchless User Interface (TUI) [10]. TUI is especially useful in case of AR & VR where we need to interact with 3D objects or real-world objects. TUI is also helpful in situations like the COVID-19 pandemic [11].

B. Gesture Taxonomy

In gesture recognition, all hand movements can be broadly classified into (i) gestures and (ii) unintentional movements [12]. Recognizing the gestures from unintentional movements is the most basic task of a gesture recognition system. The gesture taxonomy as developed by Quek [13] and further modified by Pavlovic et al. [12] can be slightly modified and represented in a better way as shown in Fig. 1.

C. Spatial Modeling of Gestures

Gestures can be modeled in 2 ways [12]: (i) 3D model based: this requires sensors like the infrared sensor, in addition to, or without a webcam. The 3D models can be either skeletal based or volumetric. Products like Microsoft Kinect and Leap Motion controller use this method. (ii) Appearance based: This uses mostly the camera to capture 2D images and then build the spatial model using smart algorithms or additional data. Mediapipe does spatial modeling of gestures using this method [15].

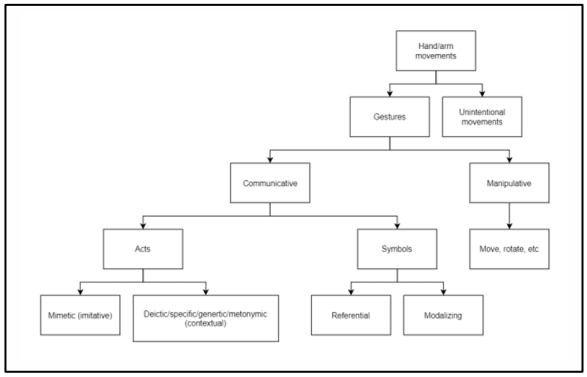

Figure 1. Gesture taxonomy

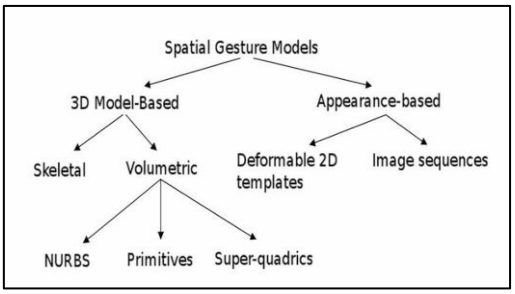

Figure 2. Spatial Gesture Modeling (source: [14])

III. DESCRIPTION

D. Application Architecture

The work consists of a Main window (class) which has GUI and visible to the user. It offers several options, including enabling/disabling of the gesture recognition software. Main window has 4 tabs: (i) Home tab: Toggle all gestures, toggle mouse gestures, toggle preview of gesture recognition. (ii) Gestures tab: displays all the gestures and helps to preview them. (iii) Help tab: a section on how to use the app. (iv) Settings tab: Change the action performed by any of the gestures. This is just the skeleton of the prototype app. The backend of the app is a separate class called Core app. It handles the gesture recognition task. It uses threading and runs in the background.

E. Gesture recognition

Gesture recognition is done with the help of a framework called Mediapipe [15]. Mediapipe first detects the hands (mainly palms) using single shot detector model [15]. It then tracks them, instead of doing recognition every frame. When a hand is recognized, it further detects 21 hand landmarks (e.g. index finger tip, wrist point, etc.). It gives the location of each of these hand landmarks through a function. The framerate or frames per second of the hand recognition and tracking can be increased by using threading.

F. Mouse pointer recognition

Trajectory based mouse pointer control has already been proposed by K. Machanda and B. Bing [16]. Here we propose a simpler algorithm.

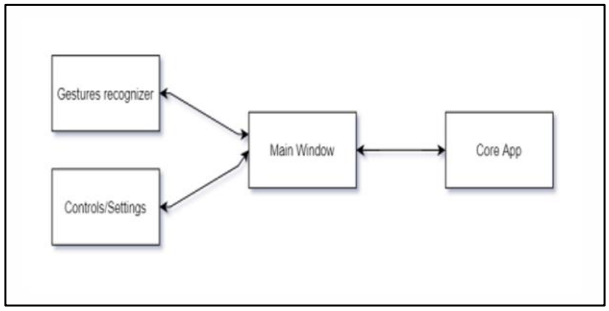

Figure 3 Architecture of the application

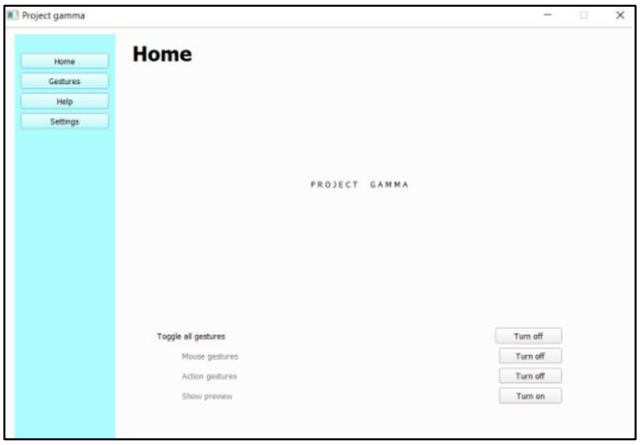

Figure 4. Application screenshot

Pseudocode:

// start of algorithm

```
Function mouse_pointer_movement():
        if only index and thumb fingers are open:
                x = curr_index_tip_x - prev_index_tip_x
                y = curr_index_tip_y - prev_index_tip_y
                x = x * (SCREEN_WIDTH / frame_WIDTH)
                y = y * (SCREEN_HEIGHT / frame_HEIGHT)
                curr_x_coordinate = prev_x_coordinate + x
                curr_y_coordinate = prev_y_coordinate + y
                move_mouse_pointer(-curr_x_coordinate, curr_y_coordinate)
                prev_x_coordinate = curr_x_coordinate
                curr_y_coordinate = curr_y_coordinate


Function for_each_frame_receive_from_webcam:
        mouse_pointer_movement()
// end of algorithm
```

As shown below, our method takes less amount of time for detection of a single frame, as compared to Lucas Kanade's method (LK) and Motion of History Images (MHI) method. Since we use the latest Mediapipe [15] framework, our accuracy (95.7% for palm detection) is much better than the other two methods, while performing better too, thanks to threading [16].

```
FPS counting method:
curr_time = time.time()
if curr_time - prev_time> 0:
        fps = 1 / (curr_time - prev_time)
else:
        fps = 0
prev_time = curr_time
```

Figure 5. Application preview screenshot

IV. MODULES IDENTIFIED

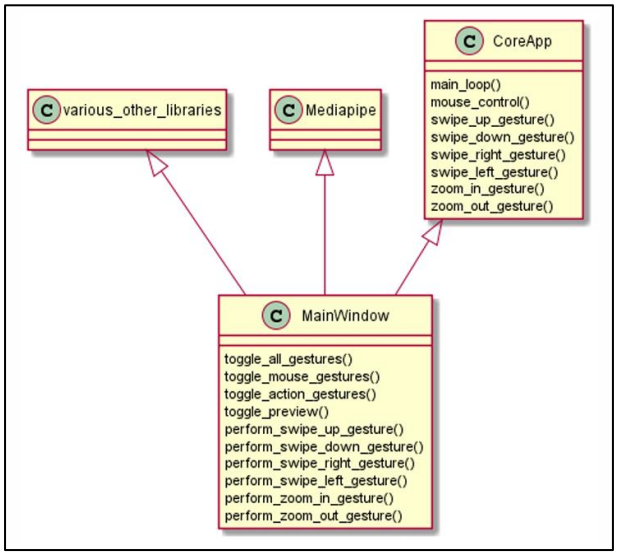

Figure 6. Modules of the application

The different classes and their dependencies has been summarized in Figure 6. The MainWindow depends on CoreApp, Mediapipe and various other libraries including PyQt5, pynput, OpenCV and PyAutoGUI.

V. RESULTS

Milliseconds required to process 1 frame:

TABLE 1. PERFORMANCE COMPARISON [16]

	LK (ms)	**MHI (ms)**	**Our method (ms)**
Accuracy	Less than MHI	70%	95.7%
Resolution 640x480	110	65	11
Resolution 320x240	75	45	14

VI. CONCLUSION

Ways of interacting with the computer haven't changed much since the boom of the personal computer itself. But computer usage has only been increasing. With the new age of AR & VR, gestures could be the main method of interaction. Hence, gesture recognition has a lot of importance in future. With this work, we aim to create a stepping stone towards the future of HCI.

VII. FUTURE ENHANCEMENT

The application can be further improved by further increasing the accuracy. More hand gestures can be added as optional features. The ability to add a custom (user defined) gesture can also be added to provide more flexibility and customizability.

REFERENCES

[1] As retrieved on 10 Nov 2021 from https://en.wikipedia.org/wiki/Minority_Report_(film)
[2] Charles Arthur, "Whatever happened to Minority Report's technology predictions?" (As retrieved on 10 Nov 2021 from https://www.theguardian.com/technology/2015/sep/18/minority-reports-technology-gestural-control-leap-motion)
[3] M. B. Khan, K. Mishra and M. A. Qadeer, "Gesture recognition using Open-CV," 2017 7th International Conference on Communication Systems and Network Technologies (CSNT), 2017, pp. 167-171, doi: 10.1109/CSNT.2017.8418531.
[4] Rico, Julie; Brewster, Stephen (2010). "Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability". Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '10. New York, NY, USA: ACM: 887–896. doi:10.1145/1753326.1753458
[5] Dalton Cooper, Here's Why Microsoft Killed Kinect, 4 Jan 2018 (As retrieved on 10 Nov 2021 from https://gamerant.com/why-microsoft-discontinue-kinect/)
[6] Lucas Matney, Once poised to kill the mouse and keyboard, Leap Motion plays its final hand, 31 May, 2019 (As retrieved on 10 Nov 2021 from https://techcrunch.com/2019/05/30/once-poised-to-kill-the-mouse-and-keyboard-leap-motion-plays-its final-hand/)
[7] Thomas Alsop, Computer penetration rate among households worldwide 2005-2019, 18 Feb, 2021. (As retrieved on 25 Apr, 2021 from https://www.statista.com/statistics/748551/worldwide-households-with-computer/)
[8] Bernard Marr, Future Predictions Of How Virtual Reality And Augmented Reality Will Reshape Our Lives, 4 Jan 2021 (As retrieved on 10 Nov 2021 from https://www.forbes.com/sites/bernardmarr/2021/06/04/future-predictions-of-how-virtual reality-and-augmented-reality-will-reshape-our-lives/)
[9] Dietrich Kammer, Mandy Keck, Georg Freitag, Markus Wacker, Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features Archived 2011-01-25 at the Wayback Machine
[10] "touchless user interface Definition from PC Magazine Encyclopedia". pcmag.com. Retrieved 10 Nov 2021.
[11] "The emerging need for touchless interaction technologies". ResearchGate. Retrieved 10 Nov 2021.
[12] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction; A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997
[13] F.K.H. Quek, "Toward a Vision-Based Hand Gesture Interface," Virtual Reality Software and Technology Conf., pp. 17-31, Aug. 1994. Retrieved on 10 Nov 2021 from https://www.researchgate.net/publication/242609555_Toward_a_Vision Based_Hand_Gesture_Interface
[14] Image by Razvan, Creative Commons Attribution 3.0 License. Downloaded on 10 Nov 2021 from https://en.wikipedia.org/wiki/File:BigDiagram2.jpg
[15] Mediapipe official documentation, https://google.github.io/mediapipe/solutions/hands.html. Retrieved 10 Nov 2021.
[16] K. Manchanda and B. Bing, "Advanced mouse pointer control using trajectory-based gesture recognition," Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp. 412-415, doi: 10.1109/SECON.2010.5453841.