Grenze International Journal of Engineering and Technology, Jan Issue

Cursor Controller using Hand Gesture Recognition

Mrudul Jain[1], Suyog Mahaganokar[2], Mrudula More[3], Rupesh Kapse[4] and Jayashree V. Bagade[5]
[1-4]Dept. of Information Technology, *Vishwakarma Institute of Information Technology*, Pune (M.H.), India
Email: mrudul.21911199@viit.ac.in, suyog.21910957@viit.ac.in, mrudula.21910499@viit.ac.in, rupesh.21910879@viit.ac.in
[5]Associate Professor, Dept. of Information Technology, Vishwakarma Institute of Information Technology, Pune (M.H.), India
Email: jayashree.bagade@viit.ac.in

***Abstract*—This paper discusses mouse cursor controller using hand gesture recognition which is an application in the field of Human Computer Interaction. The author has developed an application using the MediaPipe framework and also compared its accuracy and performance with approaches available in a variety of literature. The goal of this paper is to discuss methods to get rid of the extra hardware we need for cursor control, improve user interaction with comfortable hand gestures, reduce the computing time and increase the hit rate (accuracy/performance), as well as to use the discussed methodologies to develop software consisting of all features that a regular mouse provides. Our software must be able to perform almost all features like left Click, right-click, Brightness and volume increase/decrease, scrolling, selection, etc. By tracking the hand gestures of our user our system can perform all the above-mentioned operations with the use of a low-resolution camera only.**

***Index Terms*— Human Computer Interaction, Mediapipe, Hand gesture recognition, opencv, mouse controller, and cursor controller.**

I. INTRODUCTION

In a time when technology is heading towards the virtual world and most physical tasks being managed virtually, we are still using physical devices such as the computer mouse. The structure and features of the mouse have changed over the years, but the mouse has still remained a "hardware" device. A virtual (software-based) mouse is the need of the hour, and one that is multi-platform. At a time when the world is transitioning towards online teaching, video conferencing and online gaming etc. a virtual mouse will be helpful for increasing human-computer interaction. A virtual mouse is an application that allows users to perform mouse functionalities without using an actual physical mouse but by using hand gestures. Generally virtual mouse is operated by using other input devices like fingertip, sticks but we aim to control it using only hand gestures. While designing this virtual mouse we would try to figure out the challenges and drawbacks of hand gesture recognition, and find solutions to them. We will be using mediapipe which is google opensource framework for hand detection and opencv for image processing. In this paper authors are using Euclidean distance formula for calculating hypotenuse between two vectors which will be the basis for performing multiple functions. Frames of hands are collected by using a web camera on which different image processing functions are applied to give output frames. Stereo Vision Technology based hand gesture controllers are now the reason for increasing business in gaming industries [1]. This virtual mouse can also help in increasing business by replacing physical devices with direct hand gesture controllers. Video game sales are boosted tremendously due to motion controllers and it is

Grenze ID: 01.GIJET.9.1.558
© Grenze Scientific Society, 2023

also called the future of video games. For interaction with computers HCI is very perceptive and adequate and provides a Natural User Interface (NUI).

II. LITERATURE SURVEY

Horatiu-StefanGrif et.al [2] in their research paper, titled "Mouse Cursor Control System Based on Hand Gesture", discuss how the advent of relatively low-cost sensors (webcams) and new software packages, that offer functionalities to handle communications between computers and input devices, has helped scientists build more naturally intuitive user interfaces and opened the market for further research & investment in human computer interaction (HCI) based technologies. The paper sheds light on existing HCI devices and architectures being used in the world while presenting a case for hand-gesture based controls. The authors state since "human hand postures and gestures are still a powerful interhuman communication modality", interpolating them into computer recognizable software would help reduce inconveniences faced by new learners on unnatural input devices such as keyboards. The authors then go on to discuss a system architecture that would make the integration possible, propose hand postures to mimic a peripheral mouse device, and discuss a model that would recognize the hand postures. In their model, the authors acquire an RGB image from their webcam, convert it into an HSV image, apply a threshold operation to the HSV image and compare it to the minimum threshold hue value to see if the pixel corresponds to the hand image or the background image. For the recognition of the hand gestures, the authors consider a hand angle feature which measures three points (pixels): the highest, the extreme left and the extreme right pixels of the hand and the angle between the line segments joining these points. The authors list their results in the paper and jot down the advantages and disadvantages of their methods; they conclude by describing the improvements of their work on existing works such as [3][4]

Horatiu-StefanGrif et.al's methods are designed for static hand images and work well in conditions where the tint of background of the background is drastically different than that of the hand which isn't possible in real world scenarios. Therefore, we require a different approach for our proposed solution.

Unlike traditional mouse, a device which more naturally resembles hand-gesture based control is the Wiimote [5]. Wiimote is a peripheral device whose main purpose is to act as a controller which provides inputs for the Wii Gaming console designed by Nintendo. It has built in accelerometers and infrared (IR) sensors for providing precise and extremely receptive motion capturing abilities. The console interacts with the Wiimote using wireless technology (Bluetooth). Multiple usages of this technology were proposed, especially in the teaching and learning sector. One of the proposals was creating a system consisting of 3 main blocks: The Wiimote-based multi-touch teaching station, the Wiimote-based interactive whiteboard and the Wiimote-based stylus input conversion tool.

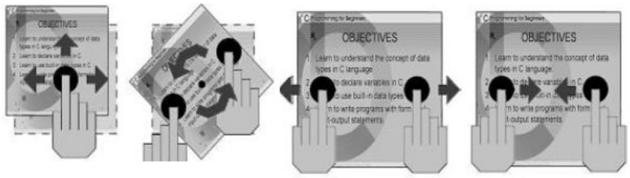

Figure 1. Basic gesture-interactions supported by the multi-touch teaching module

Even though the WiiMote supports basic gesture recognition, the use of a hand gesture to access the monitor screen from a distance is not possible and the costs associated with implementing the whole system amount to $757 which is too expensive for our use case. [6]

Therefore, market realities have proven that there still exists a scope for cheap HCI based gesture recognition solutions without the need of expensive hardware

Dipankar Gupta et.al in their research paper, titled "An Interactive Computer System with Gesture-Based Mouse and Keyboard" [7] proposed a novel multi-modal interactive keyboard and mouse system which works by taking gesture-based inputs from the user's webcam and mapping them to an onscreen keyboard. The authors of the paper achieved this functionality by using color recognition techniques in a unique manner. The user has to wrap their finger using a colored cap on their fingertip. The model then uses a computer-vision based colour recognition to identify the colour on the user's fingertip, once identified, the fingertip will act as the virtual mouse. The authors have provided 2 modules for the user to choose from: a) Keyboard module b) Mouse controlling system which can be selected using the GUI provided by the authors. The accuracy of their system is 79.17**%.** The authors then conclude the paper by listing down the advantages of the proposed technology.

Another cursor control system with the aim of achieving human computer interaction (HCI) through hand gesture detection was proposed [8]. The design consisted of creating a real time mouse controller which would

detect and analyze static hand images provided by the user and perform the corresponding action on the computer screen.

Through the medium of this dissertation, the authors have shed light on the methods they have used for recognition of hand gestures, namely, PHOG, Improved LBP and K-NN and how the architecture they used to combine their model with real time video data for the implementation of a "virtual" mouse.

The authors introduce the readers to their research topic by giving us a brief overview of the current trends in the HCI (Human Computer Interaction) domain. They go on to say that current HCI devices such as mouse, keyboards and remote controls have multiple limitations (unnatural for new users, reliance on a flat surface, limited using cables etc) which can be improved upon through the recent advances in HCI-based research, specifically, hand-gesture recognition software.

Through their research of the domain, they have concluded that there are 2 main stages of development hand gesture-based recognition can be done through: Using wearable equipment or using computer vision-based gesture recognition. Current vision-based gesture recognition methods have faced some inherent problems such as : working with images of taken as inconsistent angles, colours, scales; these issues ensured the reduction of the gesture recognition rate.

To combat these issues, the authors have proposed a method of combining PHOG features with improved LBP features to describe gestures features of the current video frame. A KNN classifier would then be used to identify the gesture by drawing a comparison of the distance between the target and actual feature vector

At this stage of the paper, the authors start describing their methodology in-depth. They have used a camera to capture video streams from the users at a frame rate of 25 fps. To improve performance, a frame is grabbed every 5 frames for processing. Each captured frame is pre-processed by converting it from the RGB format to the YCbCr format and median filtering. The conversion from RGB to YCbCr is done as it can separate brightness information from other information, and has a good result on skin color clustering. Skin color is then detected by comparing the image with threshold values of $77 < Cb < 127$ & $133 < Cr < 173$

If a skin colour is detected then the model moves onto the palm/hand detection phase. For this phase, the PHOG feature descriptor is used. HOG feature is quite commonly used for object detection due to its superior edge and contour feature separation properties. HOG is able to do this so efficiently as it takes a small area of the image and constructs the feature vector by calculating the statistical histogram of the gradient direction within this small chunk of the image. PHOG is supplemented by the improved LBP (Local Binary Pattern) algorithm which is used to describe the local texture feature information of the images. After computing the values for the HOG and LBP Feature extractors, KNN is used as a feature classifier which classifies the image into six gestures based on the data received by the feature extraction phase. KNN works by calculating the Euclidean distance between the target and actual feature vector, then after the calculations are performed, the top "K" features with the least distance are selected (as they resemble the template database the closest). Then one of the 6 categories are selected based on the frequency of features detected within that category.

The project was implemented in MATLAB combined with java.awt.Robot class (which is imported from an external library) to position the mouse cursor properly using a grid of (x,y) co-ordinates on the computer's screen. To test the accuracy of the model, the KNN model was applied to detect 6 different gestures 50 times each. Variable conditions were introduced by inputting images shot on different angles, with different lighting conditions and distorted skin tones. A common drawback of the KNN algorithm is the selection of the "K" value. Since the data was very less, the authors used K-Fold Cross Validation techniques to end up with the value of $K = 3$

On the aforementioned dataset, the model is able to get an accuracy of 86% (258/300 images properly recognized)

III. Methodology Used

In this project, we are using mainly two resources: Media pipe and Opencv. Mediapipe is a framework used for hand gesture detection and tracking. Opencv library is used for computer vision applications.

A. Mediapipe

It an open-source framework provided by Google. It is cross-platform as it is based on a time-series model. It can be applied to many videos and audio inputs and therefore is considered multimodal. Mediapipe's pipeline perception is called a graph. Nodes of media pipe graph work as calculators and edges act as a data stream. It is dependent on Opencv for the input source. Hand is represented using a 3d model but for simplification we need to convert into 2d model, which can be done with help of mediapipe. Advantages of using it as compared to other frameworks are that: 1) It requires minimum resources for processing input data, multi devices support as it

can work on many software with same performance and speed 2) once we build a model using mediapipe we can deploy it on cross-platforms such as cloud, IOS, Android, web. 3) Very user-friendly 4) Instant processing. Mediapipe helps to run heavy models with same accuracy as well as keeps pipeline lightweight and synchronized. Mediapipe uses a single-shot detector for detection of hand gestures. Single-shot detection model is very efficient in which we first take one image to detect multiple objects present in that image. The model is pre-trained and hence can process multiple output images at once. Hand landmark is mediapipe function that helps to detect multiple hands in images. In our application, we are differentiating between two hands - left and right, also minimum detection and tracking confidence is also calculated. By calculating many properties, it will give output of in the form of landmarks of palm. Media pipe possesses a self-learning feature; therefore, we can use it in low lighting conditions and blurry images, its accuracy keeps increasing with iterative processing. All these features make Mediapipe one of the best opensource frameworks compared to other frameworks.

To detect the palm and after the palm detection over the whole image our subsequent hand landmark model performs precise key point localization of 21 3D hand-knuckle coordinates inside the detected hand regions via regression, that is direct coordinate prediction. The model learns a consistent internal hand pose representation and is robust even to partially visible hands and self-occlusions [9].

B. OpenCV

Opencv is a multi-platform library for computer vision and image processing. It is open source as well as free to use. Common applications of Opencv are to remove watermarks, reduce noise, object and face recognition, drawing on images, videos, etc. It is known for its ease of processing 2D and 3D images using it. can processed using in gesture recognition after inputting images we have to covert BGR image to RGB which can be easily done by opencv also for drawing landmarks on processed image of palm it can be used. Mediapipe and opencv work explicitly to give proposed output with great performance and accuracy.

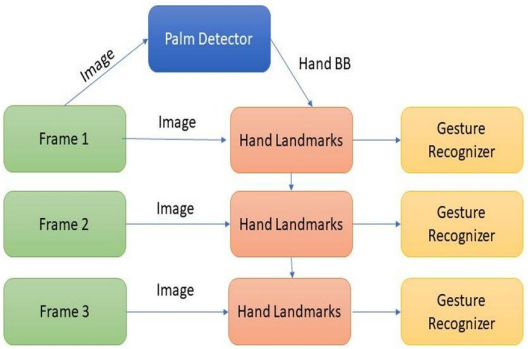

Figure 2.Mediapipe Pipeline Overview

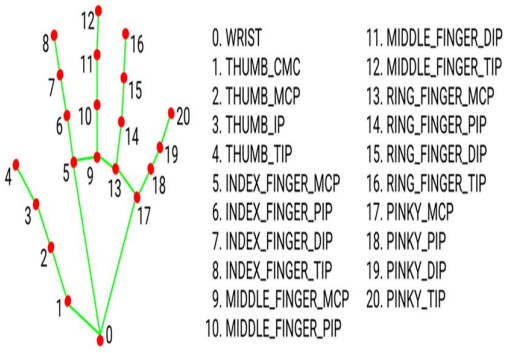

Figure 3.Handlandmarks [10] (Mediapipe Encoded)

C. Functionalities

Initially, we import libraries like mpHands, mpDraw which are mediapipe hand detection and pose detection modules respectively. Draw_utils module includes some useful functions to draw detection and landmarks over the images[11]. Opencv is used to capture images. We then convert them from BGR format to RGB using cvtColour function. The landmarks estimated in each of the detected hands are found on multi_hand_landmarks. There are a total of 21 points ranging from 0 to 20 are given to various parts of palm and by using distance and depth between them we will perform functionalities. It will give assigned flag to the hand for left its 0 and right its 1. We store minimum and maximum volume values by using pointers and ctypes. The hand labels are classified using the Library.hands function. After all landmarks get mapped, we'll apply Euclidean distance to calculate hypotenuse between the landmarks and further, use the distance to implement functions such as : volume & brightness control , scroll down, scroll up etc.

IV. RESULTS

Cross comparison of the testing of the AI virtual mouse system is difficult because only limited numbers of datasets are available. The hand gestures and fingertip detection have been tested in various illumination conditions and also been tested with different distances from the webcam for tracking of the hand gesture and hand tip detection. An experimental test has been conducted to summarize the results shown in Table I. The test was performed 25 times by 4 persons resulting in 600 gestures with manual labelling, and this test has been

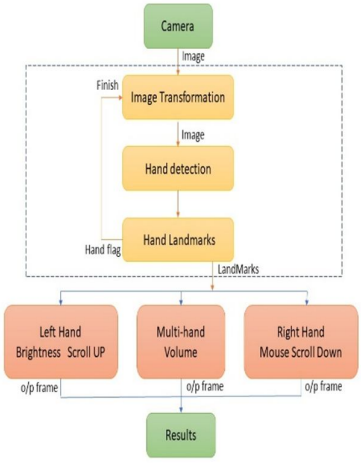

Figure 4. Block Diagram of proposed system

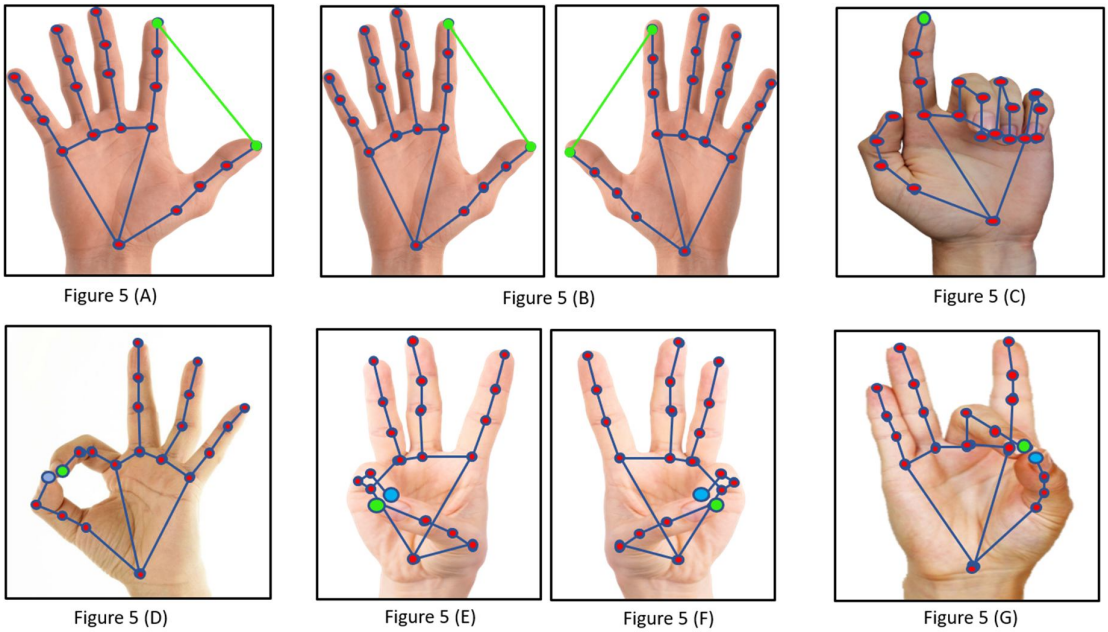

Figure 5: Hand gestures detected by the camera (Refer Table I.)

made in different light conditions and at different distances from the screen, and each person tested the AI virtual mouse system 10 times in normal light conditions, 5 times in faint light conditions, 5 times in close distance from the webcam, and 5 times in long distance from the webcam, and the experimental results are tabulated in Table II.

V. Conclusion

The application we have developed is able to perform multiple functionalities just like a hardware-based mouse would, that too with an accuracy of around 90% and good I/O performance. We are satisfied that the implementation of this virtual mouse will be able to replace a physical mouse. Features are user-friendly, accurate, and flexible with respect to various conditions. There is scope for improvement in the part of working radius and performance. We can also improve performance according to various lighting conditions. It is currently working in no light to bright light but it is quite less accurate in dim light so we can maintain its performance and accuracy irrespective of lighting conditions.

TABLE I. REFERENCE FOR FIGURE 5

Figure No.	Caption	Description
Figure 5 (A)	Hand gesture for Brightness Control	The user has to use the Left hand. Brightness is controlled using LEFT-THUMB_TIP and LEFT-INDEX_FINGER_TIP. The Euclidian distance between these two fingers determines and sets the value of brightness in the range of 0 to 100
Figure 5 (B)	Hand gesture for Volume Control	The user has to use both the Right and Left hand but the control is in the right hand. Volume is controlled using RIGHT-THUMB_TIP and RIGHT-INDEX_FINGER_TIP. The Euclidian distance between these two fingers determines and set the value of Volume in the range of 0 to 100.
Figure 5 (C)	Hand gesture for Mouse Control	The user has to use the Left hand. The mouse is controlled using LEFT- INDEX_FINGER_TIP. The position or coordinates of INDEX_FINGER_TIP are determined and responsible for setting the value of the mouse
Figure 5 (D)	Hand gesture for Mouse-Left Button	The user has to use the RIGHT hand. The left mouse button is clicked by using RIGHT-THUMB_TIP and RIGHT-INDEX_FINGER_TIP. If the Euclidian distance between these two fingers is less than the threshold i.e., in our case 25 then the left-mouse button is clicked.
Figure 5 (E)	Hand gesture for Scrolling Down	: For Scrolling-Down, the User has to use the Right hand. Scroll-Down is done by using RIGHT-THUMB_TIP and RIGHT-PINKY_TIP. If the Euclidian distance between these two fingers is less than the threshold i.e., in our case 25 then the page gets scroll-down.
Figure 5 (F)	Hand gesture for Scrolling Up	For Scrolling-Up, the User has to use the Left hand. Scroll-Up is done by using LEFT-THUMB_TIP and LEFT-PINKY_TIP. If the Euclidian distance between these two fingers is less than the threshold i.e., in our case 25 then the page gets scroll-up.
Figure 5 (G)	Hand gesture for Mouse-Right Button	The user has to use the RIGHT hand. The right mouse button is clicked by using RIGHT-THUMB_TIP and RIGHT-MIDDLE_FINGER_TIP. If the Euclidian distance between these two fingers is less than the threshold i.e., in our case 25 then the right-mouse button is clicked.

TABLE II. ACCURACY METRICS

One Finger	92%
Two Finger	88%
Three Finger	86%
Four Finger	80%
Full Hand	90%
Total	85%

TABLE III. COMPARISON OF PROPOSED METHODOLOGY WITH LITERATURE

Literature	Comparison
Horatiu-StefanGrif et.al	Horatiu-StefanGrif et.al's methods are designed for static hand images and work well in conditions where the tint of background of the background is drastically different than that of the hand which isn't possible in real world scenarios. Therefore, we require a different approach for our proposed solution.
Cheong, S.N., Yap, W.J., Chan, M.L.	In Dipankar et.al model, the user has to wrap their finger using a colored cap on their fingertip. The model then uses a computer-vision based colour recognition
Dipankar Gupta et.al	In Dipankar et.al model, the user has to wrap their finger using a colored cap on their fingertip. The model then uses a computer-vision based colour recognition to identify the colour on the user's fingertip, once identified, the fingertip will act as the virtual mouse. Our model doesn't need any such coloured finger wrapper and works only using a simple webcam. Our observed accuracy (86%) is also higher than what Dipankar et.al managed to receive (79%)
Yi, Jin & Chen, Kang & Wang, Jun & Ou, Mingmin & Wu, Yi.	Th researchers used a method of hand gesture images detection based on PHOG + Improved LBP + K-NN. They achieved an accuracy of 86%, however we instead used MediaPipe's single shot detector which even outperforms a comparable state of the art Faster R-CNN model

REFERENCES

[1] Diep, Francie. “How Gesture Control Could Transform Video Games.” LiveScience, Purch, 7 Aug. 2012, https://www.livescience.com/22157-gesture-control-could-transform-video-games.html.

[2] H.-S. Grif and T. Turc, “Human hand gesture based system for mouse cursor control,” *Procedia Manufacturing*, vol. 22, pp. 1038–1042, 2018.

[3] H.S. Grif, C.C. Farcas, „Mouse Cursor Control System Based on Hand Gesture”, Procedia Technology, 22 (2016), 657-661.

[4] H.S. Grif, Z. German, A.Gligor, Hand posture mouse. Procedia Technology, 19 (2015), 766–771

[5] Cheong, S.N., Yap, W.J., Chan, M.L. (2009). Cost-Effective Wiimote-Based Technology-Enhanced Teaching and Learning Platform. In: Muneesawang, P., Wu, F., Kumazawa, I., Roeksabutr, A., Liao, M., Tang, X. (eds) Advances in Multimedia Information Processing - PCM 2009. PCM 2009. Lecture Notes in Computer Science, vol 5879. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10467-1_112

[6] J. C. Lee, "Hacking the Nintendo Wii Remote," in IEEE Pervasive Computing, vol. 7, no. 3, pp. 39-45, July-Sept. 2008, doi: 10.1109/MPRV.2008.53.

[7] Gupta, D., Hossain, E., Hossain, M.S., Hossain, M.S., Andersson, K. (2021). An Interactive Computer System with Gesture-Based Mouse and Keyboard. In: Vasant, P., Zelinka, I., Weber, GW. (eds) Intelligent Computing and Optimization. ICO 2020. Advances in Intelligent Systems and Computing, vol 1324. Springer, Cham. https://doi.org/10.1007/978-3-030-68154-8_76

[8] Yi, Jin & Chen, Kang & Wang, Jun & Ou, Mingmin & Wu, Yi. (2020). Mouse Cursor Control Using Hand Gesture Recognition Based on PHOG-Improved LBP and K-NN Classification. Journal of Physics: Conference Series. 1576. 012048. 10.1088/1742-6596/1576/1/012048.

[9] Mediapipe hands overview. *Google Mediapipe Documentation https://google.github.io/mediapipe/solutions/hands.html.*

[10] Wu, Tung-Lin & Senda, Taishi. (2021). Pen Spinning Hand Movement Analysis Using MediaPipe Hands.

[11] Santos, Nuno. “Python: Hand Landmark Estimation with MediaPipe.” *Techtutorialsx*, 17 Apr. 2021, https://techtutorialsx.com/2021/04/10/python-hand-landmark-estimation/