Gesture-Controlled Virtual Mouse

Rahul V
[RA2111047010094]
Student
Department of CINTEL
SRM Institute of Science
And Technology
Kattankulathur, Chennai 603203

Venkaiah Chowdary
[RA2111047010104]
Student
Department of CINTEL
SRM Institute of Science
and Technology
Kattankulathur, Chennai 603203

Srijan Mishra
[RA2111047010108]
Student
Department of CINTEL
SRM Institute of Science
and Technology
Kattankulathur, Chennai 603203

Pavan Kumar
[RA2111047010099]
Student
Department of CINTEL
SRM Institute of Science
And Technology
Kattankulathur, Chennai 603203

Ananya Madhavan
[RA2111047010116]
Student
Department of CINTEL
SRM Institute of Science
And Technology
Kattankulathur, Chennai 603203

Abstract: - Gesture-controlled laptops and computers have recently gained a lot of attraction. Leap motion is the name for this technique. Waving our hand in front of our computer/laptop allows us to manage certain of its functionalities. Over slides and overheads, computer-based presentations have significant advantages. Audio, video, and even interactive programs can be used to improve presentations. Unfortunately, employing these techniques is more complicated than using slides or overheads. The speaker must operate various devices with unfamiliar controls (e.g., keyboard, mouse, VCR remote control). In the dark, these devices are difficult to see, and manipulating them causes the presentation to be disrupted. Hand gestures are the most natural and effortless manner of communicating. The camera's output will be displayed on the monitor. The concept is to use a simple camera instead of a classic or standard mouse to control mouse cursor functions. The Virtual Mouse provides an infrastructure between the user and the system using only a camera. It allows users to interface with machines without mechanical or physical devices, and even control mouse functionalities. This study presents a method for controlling the cursor's position without the need for any electronic equipment.

1. Introduction

Gesture Recognition has been a very interesting problem in the computer vision community for a long time. Hand gestures are a facet of visual communication that will be conveyed through the middle of the palm, the finger position, and therefore the shape constructed by the hand. Hand gestures are often classified into static and dynamic. As its name implies, the static gesture refers to the stable shape of the hand, whereas the dynamic gesture comprises a series of hand movements such as waving. There is a spread of hand movements within a gesture; for instance, a handshake varies from one person to a different and changes consistent with time and place. The main difference between posture and gesture is that posture focuses more on the form of the hand whereas gesture focuses on the hand movement. Computer technology has tremendously grown over the past decade and has become a necessary part of everyday life. The primary accessory for Human-Computer Interaction (HCI) is the mouse. The mouse isn't suitable for HCI in some real-world situations, like with Human Robot Interaction

(HRI). There is much research on alternative methods to the PC mouse for HCI. The most natural and intuitive technique for HCI, which is a viable replacement for the PC mouse is the utilization of hand gestures. Our vision was to develop a virtual mouse system that uses an internet camera to speak with the device in a more user-friendly way, as an alternative to employing a touch screen. To harness the full potential of a webcam, it can be used for vision-based CC, which would effectively track the hand gesture and predict the gesture based on labels. Our vision became to broaden a virtual mouse system that makes use of an internet digicam to talk with the device in a more person-friendly way, as an alternative to using a touch display screen.

2. Related Works

Implementing and realizing a virtual mouse is one prominent research area that has received considerable attention from the Human-Computer Interaction (HCI) community. The evolution of camera technology and the surge in development reduced the cost and made camera systems more affordable and prevalent than ever before. Virtual mouse applications utilize hand gestures for recognizing mouse operations. However, there exist applications where the head constitutes the main subject for the recognition. For instance, the work described in [8] offers a virtual mouse system based on head movements. In this system, the face region is detected first and then tracked during the recognition. Mouse events are realized according to the motion of the head. System robustness was evaluated based on some implicitly defined assumptions. Later, Pallejà et al. [9], developed a system whose target limb is also the head. Further, they included facial actions such as eye blinking, open mouth, and eyebrow rises in their recognition framework. However, the system suffers from catching the intended speed in response time. Xu et al. [10], investigated blink detection for their virtual mouse system. In addition, they added eye-tracking functionality which is independent from the face tracker. Their system suffered from slowness as well.

Tsang and Pun [11] proposed a virtual mouse in an embedded system based on finger tracking. For the $Y C_b C_r$ color space, they applied static threshold values offered by another study [12]. This is the same approach followed by our work described in [6], where we showed that the proposed static threshold values are not sensitive to skin but to non-skin pixels and different luminance values. Sanghi et al. [13], defined only a click mouse action for recognition; they did not conduct any performance evaluation. They offered a system based on fingertip tracking. It works only for hands on a plain white plane where a grid-based approach is followed to detect a fingertip. The system's usage as a signature input device was also presented. Yet no performance evaluation was reported.

Roh et al. [14], offered for their virtual mouse application a Bayesian network-based hand gesture recognition system. Their system operates in three steps. The first step is based on the method proposed in [15] for skin region detection. Next, the face is detected and extracted from the image. And finally, the largest remaining contour is considered to be the hand region. Afterward, fingers are identified by a set of distance-based measures, and posture estimation is achieved. Gesture recognition is accomplished by the estimated postures using a two-layered Bayesian network. For evaluation, they created their dataset as there is no publicly available dataset for virtual mouse operations. They compared their accuracy rates with other well-known methods. Although, it is declared to be real-time, unfortunately, no analysis results were reported to support this claim.

3. Methodology

The gesture-based virtual mouse system employs the transformational method to translate the coordinates of the fingertip from the camera screen to the full-screen computer window for mouse operation. When the hands are identified, a rectangular box is generated about the computer window in the camera zone, and we move the mouse pointer around the window to determine which finger is capable of completing the specified mouse operation. To create a hand-tracking, finger-only virtual mouse capable of tracking movement. Depending on the specific combination of fingers identified, we used several combinations of fingers to perform various mouse movements. The use of a virtual mouse may be observed when there is a need to conserve space or when moving around. Users of the proposed system are not required to use any gadgets or sensors, nor are they required to paint their fingertips a specific color. It enables computer-user interaction without the use of a hardware mouse device. It is also inexpensive and simple to use. The OpenCV library is used for computer vision tasks, and the Media Pipe system is used for tracking and identifying hand motions.

3.1 Modules

OpenCV: OpenCV, a computer vision package, includes techniques for processing images that detect objects. The Python computer vision package OpenCV can be used to create real-time computer vision applications. The OpenCV library is used to analyze data from photos and videos, such as face and object recognition. OpenCV is a free and open-source software library for computer vision and machine learning. With OpenCV, a standard infrastructure for computer vision applications was created to accelerate the incorporation of artificial intelligence into products. Because OpenCV is an Apache 2 licensed product, it is simple for businesses to use and modify the code.

Media Pipe: A Google open-source system called Media Pipe is used in a machine-learning pipeline. Because the Media Pipe framework was built with time series data, it can be used for cross-platform programming. Because it is multimodal, the Media Pipe architecture supports a variety of audio and video formats. The developer employs the Media Pipe framework to create and analyze systems based on graphs and also to create systems for application-related purposes. The pipeline configuration is where the system's actions in the Media Pipe are carried out. The pipeline's ability to execute on multiple platforms enables scalability on desktops and mobile devices. The Media Pipe framework is made up of three essential components: performance evaluation, a system for accessing sensor data, and a collection of reusable pieces known as calculators. A pipeline is a graph composed of units known as calculators that are linked together by streams through each data packet's pass. Developers can add, remove, or redefine custom calculators anywhere in the graph to create their applications.

PyAutoGUI: PyAutoGUI is essentially a Python software that runs on Windows, MacOS X, and Linux and allows users to mimic keyboard button presses as well as mouse cursor movements and clicks. PyAutoGUI is a Python package for cross-platform GUI automation for people. PyAutoGUI, a Python automation module, can be used to click, drag, scroll, and move objects. It can be used to precisely click where you want. It is used to automate keyboard and mouse control. In each of the three major operating systems (Windows, macOS, and Linux), there are several methods for programmatically controlling the mouse and keyboard. Complex, enigmatic, and highly technical elements are frequently involved. PyAutoGUI's role is to hide all of this complexity behind a simple API.

Math: The Python math module is a critical component designed to handle mathematical operations. It has always been present and is included with the standard Python version. The majority of the mathematical functions in the math module are simply thin wrappers for the equivalents on the C platform. Because its underlying functions are written in CPython, the math module is efficient and adheres to the C standard. Python's math module allows you to perform common and useful mathematical operations within your program. The Python math module includes a number of pre-defined constants. There are numerous benefits to having access to these constants. One advantage is that you will save time and ensure consistency throughout your entire code by avoiding the need to manually hardcode them into your program.

4. Experimental Results

We employed HCI (Human Computer Interaction) and computer vision in this work to contribute to future vision-based interactions between machines and humans. The paper in question is about using hand gestures to control mouse functions. The main tasks are mouse movement, right and left button clicks, double clicks, and up and down scrolling. There are several color bands defined, and users can choose any color from the colors that are in accordance with the backgrounds and lighting conditions. This may differ depending on one's background. For example, when a user first launches the system, he or she will be given the option of selecting a color from a palette of six options (Green, Yellow, Red, Blue, and two others). The user must select one color that does not match the background; alternatively, he must select a color that can be emphasized in the present background.

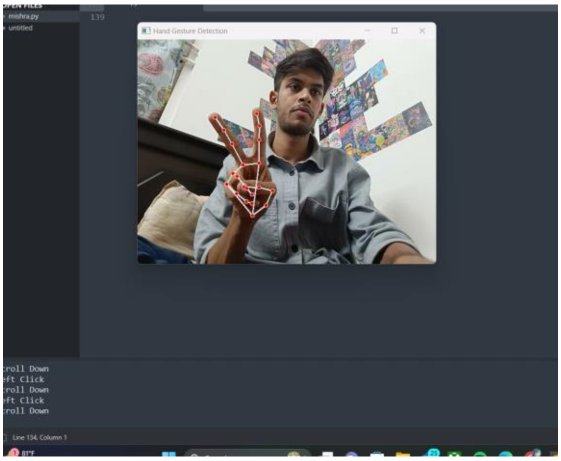

Fig 1. Neutral Click Gesture

Fig 2. Neutral Hand Gesture

Fig 3. Scanning Hand Gesture

5. Conclusion

Finally, the development of the gesture-controlled virtual mouse system, which employs object detection and gesture recognition, symbolizes a significant advancement in human-computer interaction. This study effectively demonstrated the feasibility of converting hand motions into accurate and intuitive virtual mouse actions, allowing users to control digital interfaces naturally and efficiently. The combination of YOLOv4 object detection as well as a customized CNN-based gesture recognition model, along with real-time coordination, allowed for seamless computer interaction. As this technology advances, it holds promise for a variety of applications ranging from accessibility improvements to fully immersive gaming experiences. This work paves the way for a future in which interaction between humans and computers is more intuitive and inclusive, free of the constraints of traditional input devices.

6. References

[1] https://www.researchgate.net/profile/E-Sankar-Chavali/publication/372165002_Virtual_Mouse_Using_Hand_Gesture/links/64a7b36595bbbe0c6e1c1b87/Virtual-Mouse-Using-Hand-Gesture.pdf

[2] https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e76213cc198692c58c64c30c2a4668a3bda9abdf

[3] https://easychair.org/publications/preprint_download/12Pf

[4] https://ieeexplore.ieee.org/abstract/document/9137854

[5] researchgate.net/profile/E-Sankar-Chavali/publication/372165002_Virtual_Mouse_Using_Hand_Gesture/links/64a7b36595bbbe0c6e1c1b87/Virtual-Mouse-Using-Hand-Gesture.pdf

[6] Aksaç A, Ozturk O, Ozyer T (2011) Real-time multi-objective hand posture/gesture recognition by using distance classifiers and finite state machine for virtual mouse operations. In: Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on, pages II–457. IEEE

[7] https://dl.acm.org/doi/pdf/10.1145/290747.290822

[8] Fu Y, Huang TS (2007) Head tracking driven virtual computer mouse. In: Applications of computer vision, 2007. WACV'07. IEEE Workshop on. IEEE, pp 30–30

[9] Pallejà T., Soler Edgar R., Teixidó M., Tresanchez M., Fernández del Viso A., Sánchez CR, Palacin J (2008) Using the optical flow to implement a relative virtual mouse controlled by head movements. J UCS 14(19):3127–3141

[10] Xu G, Wang Y, Feng X (2009) A robust low-cost virtual mouse based on face tracking. In: Pattern recognition, 2009. Chinese Conference on CCPR 2009, IEEE, pp 1–4

[11] Tsang W-WM, Pun K-P (2005) A finger-tracking virtual mouse realized in an embedded system. In: Intelligent signal processing and communication systems, 2005. Proceedings of 2005 International Symposium on ISPACS 2005. IEEE, pp 781–784

[12] Chai Douglas, Ngan King N (1999) Face segmentation using skin-color map in videophone applications. Circuits and Systems for Video Technology, IEEE Transactions on 9(4):551– 564

[13] Sanghi A, Arora H, Gupta K, Vats VB (2008) A fingertip detection and tracking system as a virtual mouse, a signature input device and an application selector. In: Devices, circuits and systems, 2008. 7th International Caribbean Conference on ICCDCS 2008, IEEE, pp 1–4

[14.] Roh M-C, Huh S-J, Lee S-W (2009) A virtual mouse interface based on a two-layered Bayesian network. In: 2009 Workshop on Applications of Computer Vision (WACV), IEEE, pp 1–6

[15] Brand J, Mason JS (2000) A comparative assessment of three approaches to pixel-level human skin-detection. In: Pattern Recognition, 2000. 15th International Conference on Proceedings, vol 1. IEEE, pp 1056–1059