EasyChair Preprint
№ 13263

Virtual Mouse Using Hand Gesture

D. D. Meshram, Pritish Dighore, Ayush Hedaoo, Krunal Dhoble
and Kunal Paunikar

May 13, 2024

Virtual Mouse Using Hand Gesture

Prof. D. D. Meshram[1,] Pritish Dighore[2,] Ayush Hedaoo[3,] Krunal Dhoble[4,] Kunal Paunikar[5]

[1,2,3,4,5] Department of Computer Science and Engineering, P.J.L.C.E Nagpur

Abstract:

In human-computer interaction, virtual mouse implemented with fingertip recognition and hand gesture tracking based on image in a live video is one of the studies. It is used to control the movement of the mouse pointer on the computer screen, and the other actions such as mouse move, clicks, scroll, zoom, screenshot, etc. The paper proposes a method for controlling a virtual mouse using fingertip identification and hand gesture recognition. The study introduces two approaches for tracking fingers: one involves using colour caps, and the other utilizes hand gesture detection. The process comprises three main steps: finger detection, hand gesture tracking, and implementing the tracking results on an on-screen cursor. In this research, hand gesture tracking is achieved by detecting the contour of the hand and creating a convex hull around it. Overall, this project presents an innovative approach to control the computer mouse pointer using hand gestures, which can have potential application in various fields, including virtual reality, gamming and accessibility technology.

Keywords: -Finger Tracking, Hand Gesture Recognition, Image Processing, Virtual Mouse.

1.Introduction:

Users without access to a physical mouse can still navigate their computers using a virtual mouse system, employing a standard webcam as its primary hardware component. This technology allows for the emulation of mouse functionality through hand gestures captured by the webcam. By leveraging image processing techniques, such as those provided by OpenCV, hand movements are detected and translated into mouse actions, including clicks, scrolls, and zooms. This setup eliminates the need for traditional input devices like mice and keyboards.

The virtual mouse system operates by continuously capturing video frames from the webcam and analysing them to recognize hand gestures. Once a gesture is detected, an algorithm interprets it as a specific mouse action and moves the cursor accordingly on the computer screen. Additional functionalities, such as left and right-click actions, can be implemented using PyAutoGUI, a cross-platform GUI automation library. This approach allows for intuitive and hands-free computer interaction, with potential applications extending beyond mouse emulation to include sign language recognition, database interaction, and motion control systems.

2.Literature Review:

Utilizing a hand recognition system for controlling the mouse pointer, left click, right click, drag, and other essential mouse functions within the current virtual mouse control system can provide significant benefits. However, it's crucial to acknowledge that the landscape of technological solutions is continually evolving. While the system described relies on static hand recognition, which recognizes specific hand shapes and assigns actions to each shape, it's important to explore more robust and versatile alternatives.

As technology progresses, new methods for human-computer interaction emerge. Gesture Controlled Virtual Mouse systems, for instance, integrate voice commands and hand motions to simplify computer usage while minimizing direct physical contact with the device. By combining advancements in Computer Vision and Machine Learning, these systems can recognize both static and dynamic hand movements, along with verbal commands, without the need for additional hardware.

The development of such systems often relies on cutting-edge algorithms, such as those employed by Media Pipe utilizing pybind11, to achieve the objective of providing an alternative to traditional mouse systems. Through the use of a standard web camera to capture hand gestures and movements, these systems can interpret these actions to execute various mouse operations, including left-clicks, right-clicks, and scrolling

actions.

3.Methodology:

The Methodology for Gesture Controlled Virtual Mouse incorporates the steps of Analysis, Planning, Content, Design, Programming, Testing, Marketing and Advertising

➤ Analysis: In This step, the project team would gather information about the requirement for the gesture controlled virtual mouse. This might involve researching existing products, interviewing potential users, and identifying technical requirements such as hardware and software specifications.

➤Planning: - Based on the analysis, the team would develop a plan for the project, including timelines, milestones, and resource allocation. When the project plans are documented, the project deliverables and requirements are defined, and the project schedule is created.

➤ Content: - This step involves creating the content that will be used in the product, such as graphics, user interface design, and documentation.

➤ Design: In this step, the team would create a detailed design for the product, including wireframes, prototypes, and user flows. It is an early phase of the project lifecycle where ideas, processes, resources, and deliverables are planned out. A project design comes before a project plan as it's a broad overview whereas a project plan includes more detailed information.

➤ Programming: - The actual development of the product would take place in this step, with the team writing the code to implement the design. Programming allows the project team to take the information acquired during Project Initiation and documented in the Owner's Project Requirements to the next level of detail.

➤Testing: - Once the product is developed, it would need to be thoroughly tested to ensure that it works as expected and meets the requirements that were identified in the analysis phase.

Overall, this methodology is a standard approach to software development and product promotion. However, the specific details of each step may vary depending on the nature of the product and the needs of the project team.

User selection:

Clicking:

We need to add the ability to close the object by clicking on it. This works similarly to opening it, but since there's only one object, we just need to find its center. Then, we'll move the mouse pointer to that center and click the mouse button. So instead of releasing the mouse button, we'll press it down.
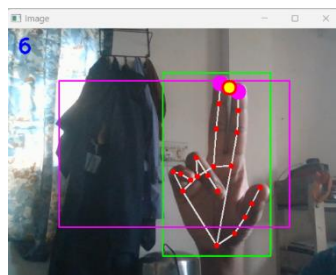

Mouse Movement:

The pink circle in the image corresponds to the index finger, serving as a marker for mouse movement. To determine the optimal location for mouse control, we employ a method to pinpoint the canter of two detected red objects. This involves calculating the average of the maximum and minimum points of their respective bounding boxes. By obtaining two coordinates representing the centres of these pink objects, we

then determine the midpoint between them. This midpoint, denoted by the pink point, serves as the reference for controlling the mouse cursor, specifically through the movement of the index finger.
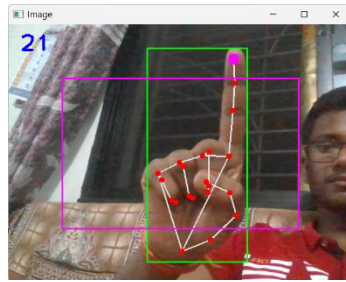

Single Click to Zoom:

With the single-click zoom feature, users can zoom in on content effortlessly using just their index finger and middle finger. By detecting a quick tap or momentary pause in movement, the system triggers the zoom function. This intuitive interaction simplifies the zooming process, enhancing user experience with minimal effort.
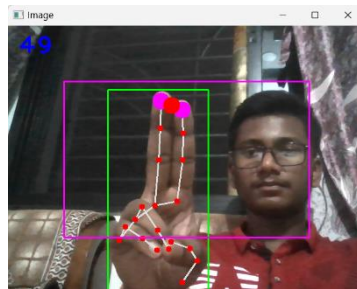

Scroll Up:

When the distance between the thumb and the tip finger increases, indicating a spreading motion, the system responds by scrolling up. This intuitive gesture allows users to navigate content upwards with ease.
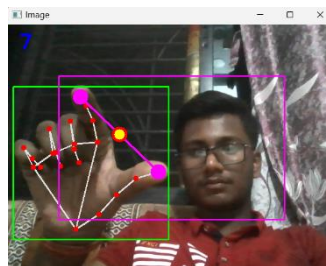

Scroll Down:

When the distance between the thumb and the tip finger decreases, suggesting a pinching motion, the system initiates a scrolling down action. This natural gesture enables users to effortlessly navigate content downwards.
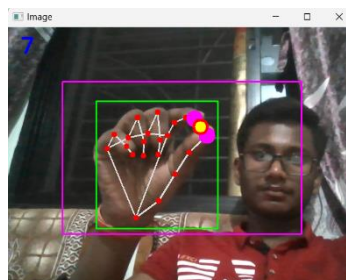

Slide Changer:

To adjust the slider to the right with a single click, users tap once on the screen. This action shifts the slider incrementally to the right, allowing for precise adjustments. Conversely, users can swipe left with three fingers on the screen to move the slider to the left. This intuitive gesture enables quick and convenient navigation of the slider controls.
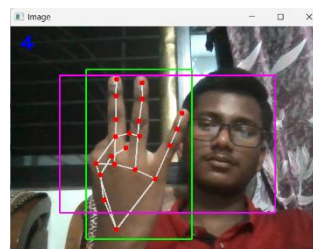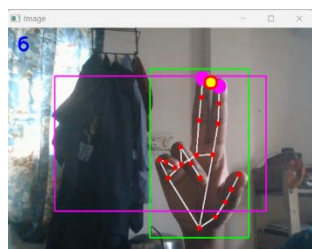
Screenshot:

Incorporating five fingers for taking a screenshot involves a straightforward process: when the system detects the spread of all five fingers on the screen simultaneously, it triggers the screenshot function. This intuitive gesture provides users with a convenient and effortless way to capture their screen contents with just a simple hand movement.
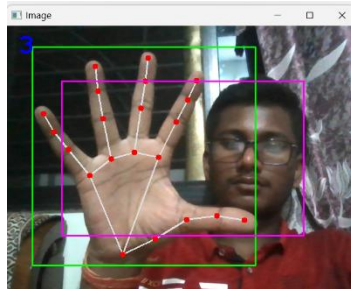
Modules:

• OpenCV

• Media Pipe

• PyAutoGUI

• Math

OpenCV: - OpenCV, a renowned computer vision library, offers a plethora of techniques for image processing and object detection. Its Python package enables the development of real-time computer vision applications, empowering users to analyze data from images and videos, including tasks like face and object recognition. This open-source software, widely known for its versatility, serves as a robust tool for both computer vision and machine learning applications. By providing a standardized infrastructure, OpenCV accelerates the integration of artificial intelligence into various products. Notably, its Apache 2 license facilitates easy adoption and modification of the codebase, making it a preferred choice for businesses seeking customizable solutions in the realm of computer vision.

MediaPipe: - MediaPipe, an open-source framework developed by Google, serves as a versatile tool within machine learning pipelines. Leveraging time series data, MediaPipe facilitates cross-platform programming, offering support for various audio and video formats due to its multimodal architecture. Developers utilize the MediaPipe framework to design and analyze systems through graph-based approaches, tailoring them to application-specific requirements.

At the core of MediaPipe lies the pipeline configuration, orchestrating actions within the system. This flexible pipeline enables scalability across desktop and mobile platforms. The framework comprises three key components: performance evaluation, sensor data access, and a collection of reusable modules known as calculators.

A MediaPipe pipeline is essentially a graph composed of interconnected calculators, forming pathways through which data packets flow. Developers have the freedom to customize the pipeline by adding, removing, or redefining calculators to suit their application needs. This flexibility empowers developers to create bespoke solutions while harnessing the power of MediaPipe framework.

PyAutoGUI: - PyAutoGUI stands as a cross-platform Python software compatible with Windows, MacOS X, and Linux, facilitating the emulation of keyboard inputs, mouse movements, and clicks. This Python package caters to GUI automation needs across various operating systems, empowering users to automate tasks such as clicking, dragging, scrolling, and precise cursor positioning.

With PyAutoGUI, users can automate keyboard and mouse controls seamlessly, abstracting away the

intricate technicalities involved in programmatically manipulating these input devices across different operating systems. Rather than grappling with complex and cryptic methods for controlling the mouse and keyboard individually in Windows, macOS, and Linux, PyAutoGUI simplifies the process through a user-friendly API, streamlining GUI automation tasks for developers and users alike.

Math: - Python's math module stands as a cornerstone for handling mathematical operations seamlessly within Python programs. This module, integral to the default Python installation, provides a comprehensive range of mathematical functions, simplifying complex calculations. Behind the scenes, many of these functions serve as thin wrappers for their C platform equivalents, ensuring efficiency and adherence to CPython standards.

A notable feature of the Python math module lies in its provision of predefined constants, offering users a repertoire of commonly used mathematical values. Leveraging these constants not only saves time by obviating the need for manual inclusion in code but also ensures consistency across the entire program.

By leveraging the Python math module, developers can execute a myriad of mathematical operations effortlessly, tapping into a robust library of functions and constants to enhance the efficiency and accuracy of their programs.

4. Result:

As technology continues to integrate into our daily routines, the realm of human-computer interaction progresses towards greater convenience. However, while these advancements are often taken for granted by many, individuals with disabilities often encounter challenges in utilizing such technologies effectively. In addressing this issue, a recent study introduces a novel gesture-based virtual mouse system that utilizes hand motions and hand tip detection. The primary objective of this proposed system is to replace traditional mouse functionality with a webcam or built-in camera, enabling users to control mouse pointer movements, scrolling, and other essential features through intuitive hand gestures. By leveraging this innovative approach, individuals with disabilities can access and interact with computers more effectively, thereby enhancing their overall user experience and inclusivity in the digital realm.

5. Conclusion:

This paper presents a novel approach for cursor control without requiring any physical sensor connections. The method involves identifying colored caps on fingertips and tracking their movement. Alternatively, different hand gestures can replace colored caps for achieving the same functionality. The mouse control functionalities include single left-click, scrolling, screenshots, right slider, left slider, zoom achieved through various combinations of colored caps or hand gestures The program allows for adjusting the range of skin colors based on the user and surrounding lighting conditions. Additionally, it analyzes the output to determine the approximate area ratio not occupied by the hand in the convex hull, enabling more precise control. This technology finds application in real-time scenarios such as cursor control on computers and Android-based smart televisions. Unlike traditional devices like mice and laser remotes, this approach minimizes the need for additional hardware by utilizing finger motions in front of a camera to perform on-screen operations.

6. References:

[1] N. Shaker and M. A. Zliekha, "Real-time Finger Tracking for Interaction," 2007 5th International Symposium on Image and Signal Processing and Analysis, Istanbul, 2007, pp.141-145.

[2] R. M. Prakash, T. Deepa, T. Gunasundari and N. Kasthuri, "Gesture recognition and fingertip detection for human computer interaction," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-4.

[3] A. Dekate, A. Kamal and K. S. Surekha, "Magic Glove – wireless hand gesture hardware controller," 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, 2014, pp. 1-4.

[4] J. Suh, M. Amjadi, I. Park and H. Yoo, "Finger motion detection glove toward human- machine interface," 2015 IEEE SENSORS, Busan, 2015, pp. 1-4.

[5] R. S. Batu, B. Yeilkaya, M. Unay and A. Akan, "Virtual Mouse Control by Webcam for the Disabled," 2018 Medical Technologies National Congress (TIPTEKNO), Magusa, 2018, pp. 1-4.

[6] A. Mhetar, B. K. Sriroop, A. G. S. Kavya, R. Nayak, R. Javali and K. V. Suma, "Virtual mouse", International Conference on Circuits, Communication, Control and Computing, Bangalore, 2014, pp. 69-72.

[7] S. K. Kang, M. Y. Nam and P. K. Rhee, "Color Based Hand and Finger Detection Technology for User Interaction," 2008 International Conference on Convergence and Hybrid Information Technology, Daejeon, 2008, pp. 229-236.

[8] Y. Fang, K. Wang, J. Cheng and H. Lu, "A Real-Time Hand Gesture Recognition Method," in 2007 International Conference on Multimedia & Expo, Beijing, 2007 pp.

[9] M. Han, J. Chen, L. Li and Y. Chang, "Visual hand gesture recognition with convolutionneural network," in 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Shanghai, China, 2016 pp. 287-291.

[10] T. D. Grove, K. D. Baker and T. N. Tan, "color-based object tracking," Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170), Brisbane, Queensland, Australia, 1998, pp. 1442-1444 vol.2.

[11] N. Vo, Q. Tran, T. B. Dinh, T. B. Dinh and Q. M. Nguyen, "An Efficient Human- Computer Interaction Framework Using Skin Color Tracking and Gesture Recognition," 2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), Hanoi, 2010 pp.