International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Cursor Movement by Hand Gesture

Abhishek Raj Mishra [a], ***Komal Kumari*** [b], ***Komal Sinha*** [a,b,*], ***Sakshi Kumari*** [a,b,c*]

[a] *First affilaton, Jain Global Campus, Bengaluru,562112, India*
DOI: https://doi.org/10.55248/gengpi.4.1123.113139

A B S T R A C T

This project introduces an advanced system for cursor movement control on a computer interface, utilizing real-time hand gesture recognition. The core technology leverages a combination of camera-based motion detection and sophisticated algorithms to interpret and translate hand movements into cursor navigation commands. This approach offers a more intuitive and natural method of interaction compared to traditional input devices like mice or trackpads. The system has been designed to recognize a variety of gestures, enabling not just basic cursor movement but also complex commands such as click, scroll, and zoom. Rigorous testing under diverse lighting and background conditions ensures the system's robustness and user-friendliness. This project not only presents a significant advancement in human-computer interaction but also holds potential for assisting individuals with mobility impairments, thereby contributing to more inclusive technology solutions.

Keywords: Collaboration, Digital Transformation, E-Learning, Innovation, Online Education, Payment Processing, Personalized Learning, Platforms, Security, Virtual Classroom Keywords: Hand gesture recognition, Cursor movement, Computer interface, Realtime motion detection, Camera-based system, Gesture interpretation, Human-computer interaction (HCI), Intuitive user experience, Accessibility, Inclusivity in technology, Gesture-based commands, Robustness under diverse conditions, Machine learning algorithms, Assistive technology for mobility impairments.

INTRODUCTION

The advancement of human-computer interaction (HCI) stands at the forefront of technological innovation, seeking to bridge the gap between digital interfaces and natural human behavior. In the realm of this evolving field, the concept of controlling cursor movement through hand gestures emerges as a groundbreaking approach. This paper delves into the development and implementation of a system that interprets hand gestures as commands for cursor control, effectively transforming the way users interact with computers. At the core of this project lies a camera-based system adept at real-time motion detection and gesture recognition. Utilizing advanced machine learning algorithms, the system is designed to accurately interpret a range of hand movements, translating them into familiar cursor functions such as clicking, scrolling, and zooming. This intuitive form of interaction not only enhances the user experience but also significantly contributes to the field of accessibility, offering an alternative mode of computer operation for individuals with mobility impairments. The pursuit of this technology is driven by the need for more inclusive and adaptable computer interfaces. As the digital world becomes increasingly integral to daily life, ensuring that technology is accessible to all is paramount. The hand gesture-based cursor movement system represents a step towards more inclusive HCI, promising a future where the barriers between human intention and digital response are further diminished.

LITERATURE SURVEY

Describe the relevant theory the two major methods for hand gesture recognition for HCI (Human Computer Interaction) are typically vision-based and hardware-based. One of the hardware-based approaches, put forth by Quam in 1990, makes use of data gloves to recognize gestures. With this technology, the user must put on a large data glove, which makes some gestures challenging. Two categories of vision based HCI exist: color marker-based approaches and color marker-less approaches. Wearing gloves or color markers is essential for the color marker-based technique, but not for the color maker-less approach. The simulated mouse operates wirelessly, enabling precise pointer control through hand gestures. It utilizes a webcam to capture images of the hand, processing these images to detect operations based on specific criteria such as the color of the hand, movement of the fingertips, and the angles formed between fingers. For any action to be recognized, a minimum angle of 15 degrees between the fingers is required. Commonly, to initiate a task, the index and middle fingers are bent at this minimum angle. The angle varies depending on the particular cursor operation being performed. As the webcam gathers input, it discerns the type of operation, subsequently triggering the corresponding cursor activity, thereby replicating the functions of a traditional physical mouse. The system detects the presence of a human hand and interprets fingertip movements to control standard mouse functions like dragging, left-clicking, right clicking, and adjusting screen views (minimizing, maximizing, zooming in, and zooming out). A specialized computer vision algorithm determines the operation type based on the object captured by the webcam. The functionality is determined by calculating the angle between fingers, particularly between the middle and index fingers. For instance, if this angle exceeds 15 degrees, a click operation is initiated. The movement of the index and middle fingers, along with the change in angle, regulates zooming functions; an increasing angle triggers a zoom-in, while a decreasing angle results in a zoom-out. The functionalities are predefined, hinging on the signals from the fingertips and their corresponding angles. The workflow of this simulated mouse application is illustrated in Figure 1, showcasing the mechanism's operation.

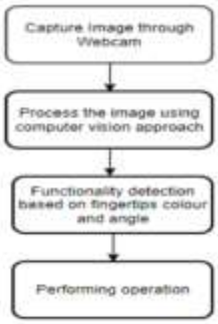

Fig. 1. Workflow of simulated mouse mechanism

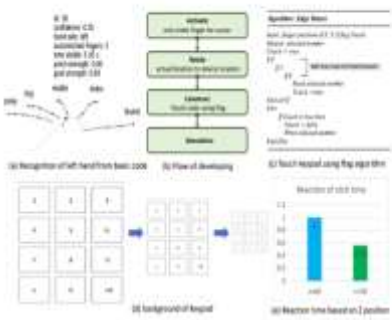

Fig. 2 Proposed method and result

The process of development hinges on a thorough comprehension of the processing language and the capabilities of the Leap Motion device. The structure, functions, tools, and variables of most processing languages bear a strong resemblance to C code. Utilizing this language, we developed basic Leap Motion code, as depicted in Fig. 2(a), enabling device operation through the analysis of hand and finger conditions. Furthermore, we have incorporated code capable of tracking finger positions and movements [3]. In our system, a cursor on a virtual keypad substitutes the traditional mouse role. To achieve this, we adapted the existing code to respond to movements of all fingers, specifically updating it to track the index finger's motion. While the original code left behind traces of movement [4], our revised version focuses solely on the current position, as outlined in Fig. 2(b). Initially, our design only allowed for lateral movement, under the assumption of a face parallel to the screen, positioned atop the device like a traditional mouse. However, considering the Leap Motion's 3D input capability, we developed new code to respond to forward and backward finger movements. Lateral movements now indicate the position of the index finger, while forward and backward movements trigger clicking actions, as explained in the button touch algorithm in Fig. 2(c). The next step involved creating a virtual keypad background, shown in Fig. 2(d), which required designing individual, functioning keys. The keypad comprises 12 squares (50x50 size), each labeled with a number. Similar to C code, the structure and sequence are crucial. Imagine painting on paper: one must first color the background before adding numbers to the keypad. Upon constructing the keypad, we calibrated the fingers' virtual location to correspond with their actual positions, dividing the keypad into sections per key. For practical applications like ATMs, where 4-digit passwords are common, recording is vital. However, our initial approach led to the repetition of the same number due to the continuous composition of the background using the for () function in C code. To overcome this, we implemented a flag system to record and react to our movements, capturing numbers as the finger shifted from front to back. To evaluate the model's accuracy, we scaled down the keypad size, as shown in the leftmost part of Fig. 2(d), and observed the effects. Despite a smaller keypad and reduced gaps between numbers, accuracy remained largely unaffected. However, this size reduction did impact the speed and caused some distortion in the response time. Interestingly, the reaction time in the x-y plane remained constant regardless of the keypad's size. A smaller keypad size led to slower movements, which, paradoxically, increased the movement distance, causing a slight offset in the program. For instance, the time taken to move the finger from number 7 to 9 remained the same in both the smallest and largest keypad sizes, as illustrated in Fig. 2. Additionally, increasing the boundary of the z-position (backward and forward) also increased the reaction time, as seen in Fig. 2(e). While performance decreased, accuracy and sensitivity were enhanced.

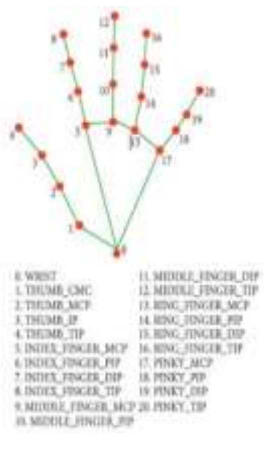

Fig 3 Movement Pointers

The system for cursor movement controlled by hand gestures operates through the analysis of frames captured by a web camera attached to a laptop or PC. This innovative system harnesses transformational algorithms to translate the finger coordinates captured on the webcam into corresponding actions on the computer's full screen, effectively controlling the mouse cursor. Key to this system is the detection and identification of fingers through the webcam. As the system discerns which finger is to execute a specific mouse function, it displays a square box on the computer window, corresponding to the area within the webcam's view where the cursor is to be maneuvered. This step involves recognizing which finger is raised by using unique fingertip identifiers, determined through Media pipe technology, along with the precise coordinates of the fingers. The mouse functions are intricately linked to hand gestures and the detection of fingertip positions using computer vision. Movement of the mouse cursor around the computer window is controlled by specific gestures: if the index finger (Tip ID = 1) is raised, or both the index (Tip ID = 1) and middle fingers (Tip ID = 2) are raised, the cursor responds accordingly. This functionality is implemented using Python's AutoPy package, which facilitates the translation of these hand gestures into smooth cursor movement across the computer's interface.

This innovative approach to cursor control represents a significant leap in human-computer interaction, enabling more intuitive and accessible use of computer systems, particularly for those seeking alternative methods of input beyond the traditional mouse and keyboard.

Proposed Methodology

Our devised methodology tackles the challenge of controlling cursor movement using hand gestures by introducing a stateof-the-art system that synergizes live image processing with progressive gesture recognition algorithms. At its core is a webcam, strategically placed and calibrated to distinguish a user's hand movements against a clear contrasting background, a critical factor for precise gesture decoding. Central to our strategy is the live image processing unit. Entrusted with a pivotal role, this unit instantaneously processes the feed from the webcam. It utilizes refined techniques to enhance the raw image, highlighting essential features like edge definition, and isolates the hand from its surroundings. This pre-processed data is crucial for the next phase, involving the gesture recognition algorithm. This algorithm represents a custom solution, drawing on the capabilities of machine learning to interpret diverse hand movements as specific cursor functions. It undergoes extensive training with a varied dataset of hand gestures, each distinctly associated with a unique cursor command. The training process delves beyond mere gesture recognition, encompassing an understanding of the subtle details, such as the angles formed by the fingers and the hand's overall posture. Post-training, the system is adept at decoding real-time gestures from the webcam, efficiently translating them into smooth and responsive cursor movements on the screen. An integral part of our methodology is the intuitive mapping of hand gestures to conventional cursor operations, ensuring the system's ease of adoption and user-friendliness. Comprehensive user testing is a critical component of our methodology, essential for validating the system's effectiveness and user satisfaction. This phase is crucial not just for assessing the system's technical performance but also for understanding the user experience. Feedback collected during this phase is invaluable, informing further enhancements to the system's ability to interpret hand gestures swiftly and accurately, even in varying environmental conditions. In conclusion, our proposed methodology provides an innovative and effective solution for hand gesturebased cursor control. By combining real-time image processing with a sophisticated gesture recognition algorithm, we aim to redefine human-computer interaction, making it more intuitive and adaptable to a broad spectrum of user requirements. Our devised methodology tackles the challenge of controlling cursor movement using hand gestures by introducing a state-of-the-art system that synergizes live image processing with progressive gesture recognition algorithms. At its core is a webcam, strategically placed and calibrated to distinguish a user's hand movements against a clear contrasting background, a critical factor for precise gesture decoding. Central to our strategy is the live image processing unit. Entrusted with a pivotal role, this unit instantaneously processes the feed from the webcam. It utilizes refined techniques to enhance the raw image, highlighting essential features like edge definition, and isolates the hand from its surroundings. This pre-processed data is crucial for the next phase, involving the gesture recognition algorithm. This algorithm represents a custom solution, drawing on the capabilities of machine learning to interpret diverse hand movements as specific cursor functions. It undergoes extensive training with a varied dataset of hand gestures, each distinctly associated with a unique cursor command. The training process delves beyond mere gesture recognition, encompassing an understanding of the subtle details, such as the angles formed by the fingers and the hand's overall posture. Post-training, the system is adept at decoding real-time gestures from the webcam, efficiently translating them into smooth and responsive cursor movements on the screen. An integral part of our methodology is the intuitive mapping of hand gestures to conventional cursor operations, ensuring the system's ease of adoption and user-friendliness. Comprehensive user testing is a critical component of our methodology, essential for validating the system's effectiveness and user satisfaction. This phase is crucial not just for assessing the system's technical performance but also for understanding the user experience. Feedback collected during this phase is invaluable, informing further enhancements to the system's ability to interpret hand gestures swiftly and accurately, even in varying environmental conditions. (1)

4. Implementation Based Approaches

The system relies on vision-based approaches, requiring no additional hardware beyond a standard camera for capturing images necessary for natural human-computer interaction accepted, be present at the scheduled time for the class.

4.1 Instrumented/Data Glove Approach

This approach uses sensor devices in a data glove to track hand movements, aiding in the precise localization and movement recognition of the hand and fingers.

Fig 3 Instrumented /Data Glove Approaches

4.2 Colored Markers Approaches

In this method, gloves with colored markers are worn to facilitate hand monitoring and location tracking of the palm and fingers.

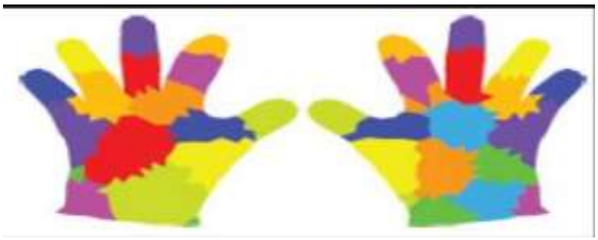

Fig 4 Colored Markers Approaches

4.3 Working Method

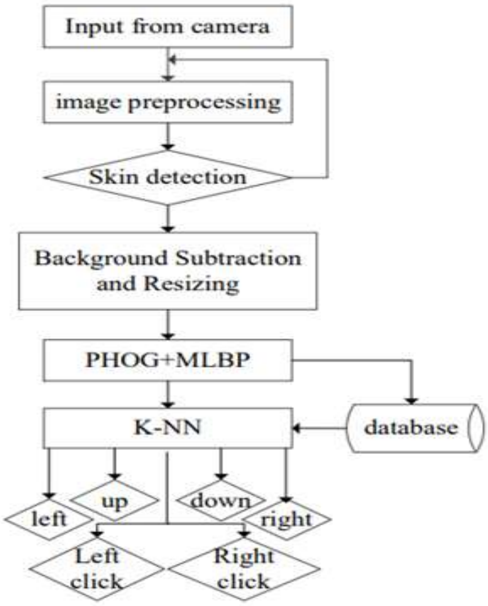

Fig 5 System Flow Diagram

The system processes video in real-time, substituting traditional mouse input with hand gesture control for a more natural user interface. Its basic block diagram is depicted in the accompanying figure. As the camera detects the palm, it captures images for processing. This includes detecting the palm, its centers, and edges, which are then used for further processing such as gesture recognition. The flowchart represents the continuous processing of each frame captured by the webcam.

5. CONCLUSION

Methods for skin detection, particularly those based on histogram analysis and explicit thresholds, were assessed for their accuracy. From these evaluations, the histogram-based method emerged as the more precise option. We developed a vision-based cursor control system utilizing hand gestures, crafted in C++ with the OpenCV library. This system successfully controls cursor movement by tracking the user's hand, executing various functions through different hand gestures. Although it shows promise as an alternative to traditional computer mice, certain limitations prevent it from being a complete replacement. One primary constraint is the system's requirement for well-lit environments, a significant drawback given the frequent use of computers in outdoor or poorly-lit settings. Enhancing the accuracy of hand gesture recognition is possible by combining template matching with machine learning classifiers. However, this approach demands a more extended implementation period, which could lead to better gesture recognition accuracy. A notable challenge encountered was the difficulty in achieving precise cursor control due to cursor instability. Incorporating a Kalman filter into the design could potentially enhance cursor stability. However, the implementation of such a filter is time-consuming, and due to project constraints, it was not feasible in this iteration. Despite these limitations, the system effectively executed all intended operations using various hand motions, yielding satisfactory results.

References

1. http://research.microsoft.com/enus/um/people/awf/bmvc02/project.pdf focuses on blood donation, specifically a life-saving blood donation application. This innovative work by Hamlin, M. R. A., and Mayan, J. A. (2016, 16-17 Dec. 2016) represents a significant contribution in this field.
2. In the work of Vijay Kumar Sharma, titled "Virtual Mouse Using Hand Gesture," published in the GIS Science Journal on December 29, 2020, the study explores the development of a virtual mouse controlled through hand gestures, highlighting the advancements in interactive technology.
3. The research by Amardeep Ghodichor and Binitha Chirakattu, "Virtual Mouse using Hand Gesture and Colour Detection," published in October 2015 (Volume 128 - No.11), alongside Chorea P., Paliwal G., Badhan P., 2013's work on "Image Processing Based Colour Detection" in the International Journal of Emerging Technology and Advanced Engineering (Volume 3, Issue 4, pp. 410- 415), delves into the innovative use of colour detection and hand gestures for virtual mouse control.
4. Rhitivij Parasher and Preksha Pareek's study, “Event Triggering Using Hand Gesture Using Open CV,” published in February 2016 (Volume -02, pages 15673-15676), investigates the application of hand gestures in event triggering, utilizing the Open CV framework.
5. The research by Dalal N. and Triggs B., “Histograms of Oriented Gradients for Human Detection,” presented at the Computer Vision and Pattern Recognition Conference in 2005 (CVPR 2005, IEEE, pp. 886-893), focuses on human detection techniques using gradient histograms.
6. The work by H. Lategahu, S. Gross, and T. Stehle, et al., titled “Texture Classification by Modeling Joint Distributions of Local Patterns with Gaussian Mixtures,” published in the IEEE Transactions on Image Processing (Vol. 19, No. 6, pp. 1548-1557, 2010), explores texture classification through the lens of local pattern modelling.
7. Chien-Cheng Lee, Cheng-Yuan Shih, and BorShennJeng's 2010 paper, “Fingertip-writing Alphanumeric Character Recognition for Visionbased Human Computer Interaction,” presented at the International Conference on Broadband, Wireless Computing, Communication and Applications, delves into vision-based HCI through fingertip character recognition.
8. In the study “Cursor Movement in Real-Time using Hand Gesture Recognition” by T. Suganya Thevi and C. Chitra, published in the International Journal of Engineering Sciences & Research Technology (ISSN: 2277-9655) in March 2014, the focus is on real-time hand gesture recognition for cursor movement.
9. The research by Shany Jophin, Sheethal M.S, Priya Philip, T M Bhruguram, “Gesture Based Interface Using Motion and Image Comparison,” found in the International Journal of Advanced Information Technology (IJAIT, Vol. 2, No.3, June 2012), investigates the use of gesture-based interfaces employing motion and image comparison techniques.
10. “Mouse Control using a Web Camera based on Colour Detection” by Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj, Hemanta Saik, published in the International Journal of Computer Trends and Technology (IJCTT, volume 9 number 1, ISSN: 2231-2803, March 2014), examines the control of a mouse using a web camera, focusing on color detection methods.