Optimanus Synaptic Interface: Advanced Cursor Control with Multimodal Integration

Prof. Madhav Ingle,Shiv Tarsode,Sarthak Jangade, Naresh Vasave

Department of Computer Engineering JSCOE, Pune
Pune, India

.

ABSTRACT;- The Optimanus Synaptic Interface (OSI) is a breakthrough in how people can interact with computers. It's a system that uses multiple types of inputs to control a cursor, giving users more ways to interact with digital devices beyond traditional tools like a mouse or touchscreen, which can feel limited and less adaptable. OSI combines different input methods, including eye movement tracking, brain signals, hand gestures, and voice commands, to create a more natural and smooth experience for users. By using advanced machine learning, especially Convolutional Neural Networks (CNNs), OSI processes these inputs instantly, accurately translating them into cursor movements

This flexible system can be used in a range of areas, making it especially useful in virtual and augmented reality, assistive technology for people with disabilities, and specialized tools for professionals in design, gaming, and remote work. Initial testing shows that OSI is easy to use and responds quickly, allowing users to achieve more precise control. These results suggest that OSI could change the way people interact with digital devices, making the experience more immersive, accessible, and intuitive.

Keywords: Machine Learning, Optimanus Synaptic Interface (OSI), Hand Gesture Recognition, Brain-Computer Interface, Voice Command, Convolutional Neural Network (CNN), Multimodal Interface, Human-Computer Interaction

I. INTRODUCTION

As digital technology becomes a bigger part of our daily lives, how we interact with computers and devices is evolving too. For years, we've relied on standard tools like keyboards, mice, and, more recently, touchscreens to control our devices. While these tools work well enough, they also have their limits. Traditional input devices often require physical effort, may not always feel precise, and can lack the natural flow we want when engaging with digital content. In situations where hands-free control is ideal or where someone's physical abilities make using these devices challenging, there's a clear need for more advanced and flexible ways to interact with computers. This is where the Optimanus Synaptic Interface (OSI) comes in.

The OSI system is designed to make digital interaction smoother and more intuitive by bringing together multiple ways to control the computer — such as eye tracking, brain signals, hand gestures, and voice commands — all in one. Rather than relying only on a mouse or keyboard, OSI lets people control a computer in more natural ways, using their gaze, thoughts, gestures, or voice to interact with digital environments. The idea behind OSI is to create an experience that feels almost effortless, pushing the boundaries of what's possible in human-computer interaction.

The inspiration for OSI comes from major advances in areas like machine learning, artificial intelligence, and neuroscience. As we better understand how the human brain and body work, we're able to develop technology that can more closely adapt to what we're thinking and doing. For example, eye- tracking technology has become so precise that a person can move a cursor just by looking at different points on the screen. Similarly, brain-computer interfaces (BCIs) have come a long way, making it possible to control basic commands using brain signals, which is perfect for hands-free interaction. Gesture recognition has also improved thanks to better computer vision and AI, so users can click, scroll, or drag items simply by moving their hands.

Machine learning is the engine that powers OSI's ability to process these various inputs in real-time. Convolutional Neural Networks (CNNs), which are particularly good at interpreting visual data, help OSI recognize hand gestures and eye movements quickly and accurately. This real-time processing is essential for the interface to feel responsive and for actions to happen instantly as the user intends.

The multimodal design of OSI — which integrates eye tracking, BCI signals, hand gestures, and voice commands — makes it adaptable to a wide range of situations. For instance, in virtual and augmented reality (VR/AR), where traditional input devices can feel awkward, OSI provides a natural way to interact with 3D environments without physical controllers. For people with disabilities, OSI's brain-computer and voice command options create new ways to use digital devices, making technology more accessible and usable. In fields like design, gaming, and remote collaboration, OSI offers a new level of precision and control. Designers can use hand gestures and eye tracking to manipulate digital objects with more freedom, gamers can interact more naturally in virtual worlds, and teams working remotely can navigate shared digital spaces more fluidly.

Even though OSI shows a lot of potential, there are still some challenges. Integrating multiple input types requires careful calibration to maintain accuracy, and some minor delays or occasional calibration issues can come up, especially during more complex tasks. Improving responsiveness and making calibration smoother and more adaptive to each user's behavior are ongoing goals for OSI's development.

Another challenge is getting the right balance between user control and system adaptability. OSI needs to interpret signals from multiple sources — eye movements, brain signals, hand gestures, and voice commands — all at once. To ensure that it accurately understands the user's intentions, machine learning algorithms need to be trained to filter out accidental movements or unintended commands. Achieving a balance where the system feels responsive and reliable is essential to making OSI feel truly intuitive.

Initial tests of OSI have shown encouraging results. Users have reported that OSI feels easy to use and intuitive, particularly in scenarios where hands-free control is useful. Feedback has highlighted that the combination of eye tracking and hand gestures creates a seamless experience that allows users to interact with digital spaces naturally, without the need for physical effort. In VR, assistive technology, and professional design contexts, OSI provides a level of freedom and control that goes beyond what a mouse or touchscreen can offer. However, further improvements in responsiveness, accuracy, and calibration will help OSI fully realize its potential.

In summary, the Optimanus Synaptic Interface is a significant step forward in human-computer interaction, aiming to make digital interactions more accessible, natural, and immersive. By combining multiple input methods and using machine learning to interpret them in real-time, OSI has the potential to transform a range of fields by offering a more flexible and intuitive way to interact with computers. From virtual and augmented reality to assistive technology and professional applications, OSI could shape a future where interacting with digital environments feels as natural as interacting with the physical world. This paper will dive into the design and workings of OSI, share findings from early tests, and explore possible applications and future improvements that could make OSI a key player in the evolution of digital interaction.

II. OBJECTIVES

This research aims to find better ways to control cursors by using the Optimanus Synaptic Interface. We plan to combine various input methods, such as brain signals, touch, voice, and gestures, to make cursor control easier and more precise. By testing how well it works and getting feedback, we hope to demonstrate how this technology can be useful in both daily use and helping people with disabilities the objectives are:

A) We want to figure out how our brains can directly control computers, like moving a cursor on a screen. We're interested in how brain signals can make cursor control more precise and even help us navigate virtual worlds.

B) Our goal is to create a system that combines different ways to input information, such as touch, voice, gestures, and brain signals. We want to make sure all these methods work together seamlessly to make cursor control more accurate and easy to use.

C) We'll test our device, the Optimanus Synaptic Interface, to see how well it performs real-world tasks. We'll compare its speed and accuracy to traditional methods like a mouse or keyboard.

D) We'll also study how using this advanced cursor control affects people mentally. We'll measure how much mental effort is needed and how the system impacts user performance through feedback and tests.

E) We're excited about the potential of this technology to help people with disabilities control devices more easily. We'll also explore itapplications in virtual reality, robotics, and healthcare.

III. METHODOLOGY

This methodology for the Optimanus Synaptic Interface project lays out the key steps and techniques used to create an advanced cursor control system that combines multiple types of input—like brain signals, touch, voice, and gestures—into a flexible, easy-to-use interface. By focusing on essential areas like real-time monitoring, data processing, algorithm use, and thorough system testing, this approach ensures the interface will work smoothly, accurately, and safely in a range of settings. Each part of the methodology builds toward a strong framework that allows users to interact naturally and reliably with the system. Altogether, this approach aims to make the Optimanus Synaptic Interface a highly responsive and adaptable tool for both everyday and assistive applications.

3.1. Real-Time Alerts and Threat Detection

Real-time alert and threat detection is essential for making sure the Optimanus Synaptic Interface stays safe and accurate. The interface works by gathering input from several different sources, including neural signals (brain waves), touch, voice, and gestures. Since it relies on multiple types of input, keeping everything synchronized and accurate is crucial for giving users a smooth experience. For example, neural signal data might vary due to changes in the environment, or gesture recognition could struggle if lighting conditions suddenly change. Real-time monitoring of these input qualities helps catch any issues before they affect the user's experience. This monitoring system continuously checks factors like signal strength, noise levels, and consistency in each input source to ensure everything stays stable.

The alert system operates by using algorithms that detect when inputs go beyond certain limits. For instance, if a brain signal's clarity falls below an acceptable level due to interference, or if the system detects frequent errors in gesture recognition, it immediately triggers an alert. This alert tells either the user or the system itself that something is wrong. Such alerts allow the interface to either recalibrate itself or, if necessary, focus on just one input type temporarily until the issue is resolved. This way, users don't have to deal with disruptions or inaccurate cursor movement, as the system adjusts to keep things running smoothly.

For assistive applications, the alert system plays an especially important role in preventing unintended actions. For example, if a user has muscle spasms that could unintentionally activate commands, the system can detect these involuntary actions and prompt the user for a double confirmation before carrying out commands. This mechanism keeps the system under the user's control, preventing accidental commands from causing frustration.

Overall, the real-time alert and detection system is a safety net that allows the Optimanus Synaptic Interface to offer reliable, user-centered functionality, particularly useful for people using assistive technology.

3.2. Proposed System

The Optimanus Synaptic Interface combines several input methods—neural, touch, voice, and gesture— into one versatile cursor control system. The main parts of this system include neural sensors to pick up brain signals, touch interfaces for manual control, voice recognition for spoken commands, and sensors for tracking gestures. All of these are connected to a central processing unit (CPU) that's responsible for combining and interpreting the inputs. With this setup, the system can switch seamlessly between different types of control based on user needs, making it flexible and responsive to different situations. For instance, if a user wants to switch from using touch to voice commands, the system can do so automatically, adapting to the user's preference in real time.

The brain sensors, such as EEG devices, capture signals that translate into simple commands. Meanwhile, the touch input offers more traditional control, with added responsiveness and sensitivity adjustments. Voice commands are understood using natural language processing, making it easier to give instructions naturally. Gesture tracking captures hand movements, giving users an even more intuitive control option. The CPU manages all these inputs and prioritizes the current input mode, making sure that transitions between control types are smooth and that the cursor responds accurately to whichever input is active.

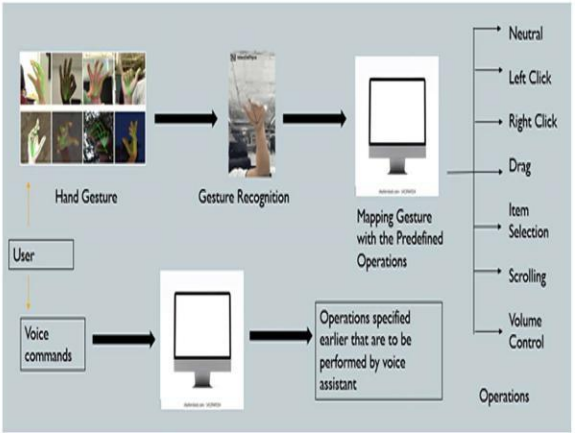

Figure 3.2

To help users stay aware of the system's status, a graphical user interface (GUI) provides real-time feedback on which input is active, cursor speed, and any alerts. This visibility makes it easier for users to adjust settings as needed and ensures they always have a clear understanding of how the interface is responding to their commands. With this design, the system can adapt to various uses, from day-to-day tasks to more specialized applications, particularly for assistive technology, where flexibility and ease of use are critical.

3.3. Data Flow

Data flow in the Optimanus Synaptic Interface starts with gathering raw input data from each modality, including brain signals, touch, voice, and gesture. This data goes through a preprocessing stage, where each type of input is refined for accuracy and noise reduction. Neural signals, for instance, may require filtering to remove any electrical interference, while voice commands go through noise reduction to isolate the spoken instructions. Each input type has its own specific processing needs, ensuring that only clear, relevant information is passed forward for use.

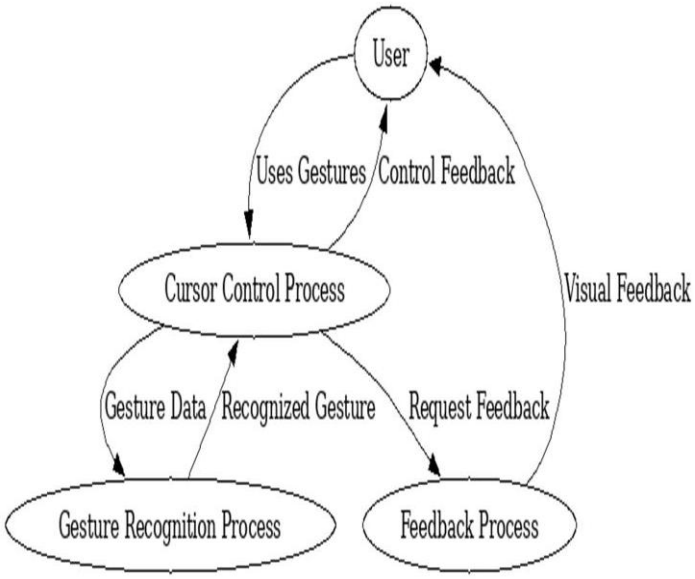

Figure 3.3

After preprocessing, the refined data reaches the multimodal integration layer. This is where the data from all input sources is combined in real time. A fusion algorithm manages this process by weighing each type of input according to user-defined priorities or context. For example, neural input might be given the most weight for moving the cursor, while gestures might serve as secondary commands for special actions. The fusion algorithm's ability to adjust dynamically allows the system to support different interaction styles, giving users a seamless, adaptable experience.

Once integrated, the decision layer interprets the combined inputs to decide on specific cursor actions, like moving in a particular direction or executing clicks. The decision layer also makes sure any conflicting inputs are resolved so that the cursor behavior is predictable. Finally, a feedback loop updates users on the current input mode, system status, and any errors, providing a way to adjust settings and ensuring full control over the interface.IoT Buzzer Alert: Simultaneously, the control room sends a signal via the NodeMCU to trigger the buzzer in the deployed area.

3.4. Algorithms for Input Processing and Data processing

The Optimanus Synaptic Interface relies on several algorithms to handle the complexity of processing and combining multiple inputs. First, signal filtering algorithms like Butterworth filters and Fast Fourier Transform (FFT) are used to clean up neural and gesture inputs, removing noise and isolating the useful signals. This preprocessing step is crucial for ensuring accurate and smooth cursor control, as it prevents distortions from affecting the input. For voice and gesture recognition, machine learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) help interpret commands accurately in real time.

Data fusion is a key part of this system, using algorithms like Kalman filters to synchronize and merge different inputs. The fusion algorithm adjusts itself according to the context; for instance, if a user is switching between touch and voice commands, the algorithm prioritizes the active mode. Decision- making algorithms then determine which input modes are prioritized, keeping command flow smooth and avoiding conflicts. These decision- making algorithms can either be rule-based or adapt based on machine learning, learning user preferences over time. Finally, an alert detection algorithm monitors input quality, using threshold or anomaly detection models to catch inconsistencies. Together, these algorithms create a reliable, adaptive cursor control system.

3.5. System Testing and Deployment

Testing and deployment ensure the Optimanus Synaptic Interface is ready for real-world use. The process starts with unit testing for each input type— neural, touch,

voice, and gesture—to verify each one works correctly on its own. Integration testing follows, checking that all parts work together without causing delays or conflicts.

Performance testing measures accuracy, speed, and reliability across various scenarios, ensuring the cursor remains consistent. This helps confirm the system's reliability under different conditions.

User testing gathers feedback on usability, including how intuitive the system is and the mental workload it requires. This feedback is essential for refining the system to create a user-friendly experience. Additionally, the alert system undergoes thorough testing to make sure it correctly identifies and signals potential input issues without causing unnecessary interruptions for the user.

After testing, the system is deployed in a controlled environment, such as a lab or specific user group, where it's monitored for any unexpected problems. During this period, the system's performance is tracked, and data is gathered to assess its functionality over time. Feedback from this phase helps in making any last adjustments, ensuring the system remains adaptable and easy to use. With this careful process, the Optimanus Synaptic Interface is prepared for reliable performance in real-world applications.

3.6 Fingertips Detection

Finger tip detection is a key component of gesture recognition within the Optimanus Synaptic Interface, enabling precise and intuitive interaction with the system. This technology focuses on identifying the position and movement of the user's fingertips in real time, which is essential for translating physical gestures into cursor control actions. By accurately detecting the fingertips, the system can interpret specific gestures such as pointing, swiping, tapping, or pinching, allowing users to control the cursor seamlessly with hand movements.

The detection process typically involves using advanced techniques such as computer vision or depth-sensing cameras. Computer vision algorithms analyze the visual input from cameras or sensors to locate the fingertips based on features such as shape, position, and contour. Depth sensors or infrared (IR) cameras can also enhance finger tip detection by providing three-dimensional data, allowing the system to track the position of the fingers in a 3D space with greater accuracy, even in varying lighting conditions or environments.

Machine learning models, including convolutional neural networks (CNNs), are often employed to improve the accuracy and adaptability of finger tip detection. These models can be trained on large datasets to recognize different

Figure 3.6

hand shapes, sizes, and finger movements, making the detection system robust across diverse users and contexts. Once the fingertips are detected, the system processes the data and converts it into commands that control the cursor, enabling a more natural, fluid interaction.

Finger tip detection is especially valuable in applications requiring fine motor control, such as selecting specific targets on a screen or performing detailed gestures in assistive technologies. It offers a hands-free and highly customizable interface, making it an essential feature of the Optimanus Synaptic Interface for both everyday and assistive applications.

IV. ARCHITECTURE DIAGRAM

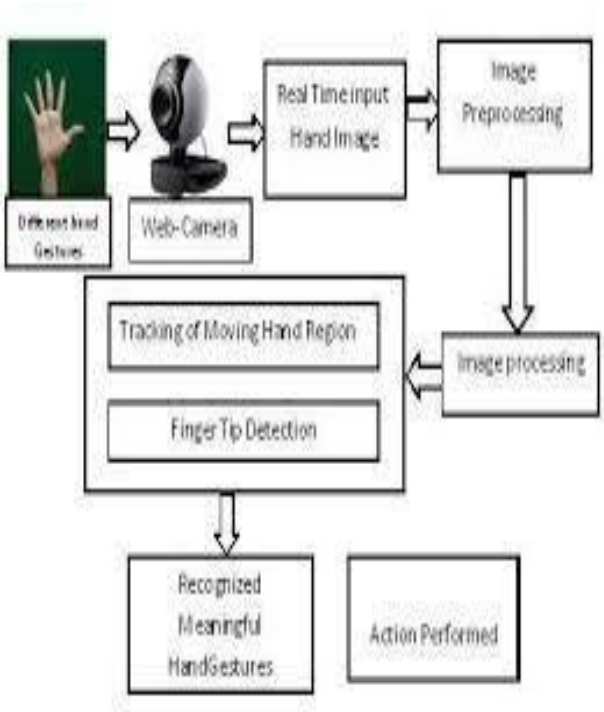

Figure.4

The "Optimanus Synaptic Interface" is an advanced system that lets users control a computer cursor without touching any device, simply by using hand gestures. It's designed to make interacting with digital screens feel natural and seamless. The system works by capturing images of a user's hand movements with a camera. These images go through several steps to clean up and focus on the hand, making it easier to detect specific gestures.

After capturing and enhancing the hand image, the system tracks the hand's position and identifies key points, like fingertips, to recognize different gestures. For example, pointing, waving, or pinching gestures can be detected and matched to commands such as moving the cursor, clicking, or zooming. When a gesture is recognized, the system performs the corresponding action on the screen.

Beyond gestures, the interface can incorporate other input methods like voice commands or even eye-tracking. This multimodal approach provides an even more flexible and intuitive experience. Overall, the Optimanus Synaptic Interface allows users to control devices with simple hand movements, creating a touch-free, interactive experience that's responsive and user-friendly.

1. Hand Gesture Capture

The first step in this system is capturing hand gestures through a web camera or similar image capture device. The camera continuously streams live video of the user's hand, serving as the primary input source. This real-time input is crucial, as it enables the system to detect and interpret hand gestures instantaneously. The camera's quality—resolution, frame rate, and sensitivity—affects the system's ability to track small, fast, or complex gestures accurately. Additionally, lighting conditions play an essential role; the system might perform additional adjustments to ensure the hand is clearly visible regardless of ambient lighting. This initial capture provides a foundation for all further processing and is optimized to work in various conditions to ensure reliable gesture recognition.

2. Image Preprocessing

After capturing the hand image, the system preprocesses it to enhance critical features while reducing noise or unnecessary details. Image preprocessing typically involves several steps:

Noise Reduction : Filters out random variations or pixel distortions, often caused by lighting inconsistencies or camera noise.

Background Subtraction: Separates the hand from the background, isolating it as the primary object of interest. This can be achieved by color segmentation, where skin color is detected and isolated, or by background modeling, where anything in motion (like a hand) is identified against a static background.

Normalization : Adjusts brightness and contrast to make the hand more prominent, especially under varying lighting conditions.

By focusing solely on the hand, preprocessing simplifies the image, enhancing the visibility of important features like fingertips and contours. This stage ensures that the hand is the only element passed on for analysis, reducing computational load and improving accuracy in later stages.

3. Image Processing

In this stage, the system performs image processing to detect and define key hand features, creating a clearer representation of the hand's structure. This involves techniques such as:

Edge Detection : Identifies boundaries of the hand, such as the edges of fingers and the outline of the palm, using algorithms like the Canny or Sobel operator. This is crucial for distinguishing the hand shape from other objects.

Morphological Operations : Sometimes used to refine the hand's shape, removing small artifacts or filling gaps within the detected hand region. This includes dilation, erosion, and other pixel-based modifications.

The output of this stage is a well-defined image of the hand, with clear edges, contours, and shapes. These enhanced features form the basis for detecting gestures in the following stages. Accurate image processing allows the system to recognize even subtle gestures and finger movements by providing a high-contrast, well-defined hand shape.

4. Hand Tracking and Feature Detection

Hand tracking is essential for monitoring the hand's position and movement over time, especially for dynamic gestures like swiping or pointing.Hand tracking involves:

Tracking Algorithms : These algorithms (such as the Kalman filter or optical flow methods) enable the system to follow the hand's position across frames in real-time. Optical flow, for instance, detects the

movement of pixels associated with the hand, allowing the system to understand both the hand's location and movement path.

Feature Points : Detects specific points of interest, such as fingertips, knuckles, and the center of the palm. This helps in recognizing gestures based on finger positioning and movements. For example, fingertip detection is crucial for distinguishing between gestures like pointing or pinching, where individual fingers play a central role.

By tracking these features in real-time, the system can monitor gestures continuously, enabling users to perform complex or multi-step gestures. This tracking also ensures accuracy, as the system can account for movements like rotating, expanding, or contracting of the hand.

5. Gesture Recognition

Gesture recognition is the stage where the system interprets hand positions and movements to identify specific gestures. It involves several processes:

Pattern Matching: This technique compares the current hand shape and finger positions against a database of predefined gestures. Each gesture has a unique pattern or "signature," which the system uses to make a match.

Machine Learning and Deep Learning: For more complex gestures, the system can use machine learning models (like decision trees, SVMs) or deep learning networks (such as CNNs) that have been trained on various hand shapes and movements. These models allow the system to recognize gestures in real-time, even when conditions vary slightly (e.g., lighting or hand angle).

Rule-Based Recognition : For simpler implementations, the system may use rule-based logic, where each gesture has a specific rule or condition (such as "fingers spread wide means open palm").

The recognized gestures are then mapped to specific commands, which enables the user to interact with the device. This stage is crucial because it determines the meaning of each gesture and translates it into an actionable command.

6. Action Execution

Once the system recognizes a gesture, it executes the corresponding **action** on the user interface. This could involve actions like moving the cursor, clicking, scrolling, zooming, or even custom commands for specific applications. The action execution stage works as follows:

Mapping Gestures to Actions : Each recognized gesture is mapped to a specific command or function. For example, a pinch gesture might zoom in, while a pointing gesture moves the cursor.

Command Execution : The system translates the recognized gesture into an input command for the computer. For example, a virtual "click" could occur when the user makes a fist, while a "drag" might happen when the user moves an open hand.

Feedback Loop : To ensure responsiveness, the system might use a feedback loop to adjust for any misinterpretation or to reset the gesture recognition if no action is performed within a set timeframe. This loop maintains system stability and improves user experience.

By associating each gesture with a corresponding command, the system offers a seamless interface, allowing the user to control the computer or device without the need for traditional input devices.

Multimodal Integration :

In addition to hand gestures, the "Optimanus Synaptic Interface" integrates other forms of input to create a more versatile and responsive interface.Multimodal integration may include:

Voice Commands : Voice recognition allows users to issue commands verbally, complementing hand gestures for a more flexible interaction method. For instance, a user might say "select" while pointing to a specific area on the screen.

Eye-Tracking : Eye-tracking detects where the user is looking on the screen, helping the system understand the user's focus area. Combining eye-tracking with gesture recognition can make interactions faster and more accurate, as the system can "guess" the intended interaction based on gaze.

Haptic Feedback : Though not part of the gesture recognition per se, haptic feedback (through vibrations or tactile responses) provides the user with confirmation or feedback, making the interaction feel more natural and complete.

This multimodal approach enables the interface to adapt to various contexts, offering flexibility and catering to different user preferences. By combining gestures with other inputs, the system creates a richer, more intuitive interaction experience.

V. Result and Analysis

This system is a Advanced Cursor Control with Multimodal Integration" integrates hand gestures and voice commands as primary input modalities for controlling a cursor. Initial testing revealed that the hand gesture control system performed well in detecting specific gestures, allowing users to move the cursor with a high degree of accuracy. However, some users found it difficult to maintain precision during more intricate tasks, suggesting that gesture sensitivity and recognition algorithms need further refinement to improve fine control. Additionally, while the hand gesture system worked reliably in controlled environments, it showed reduced accuracy when users performed rapid or complex gestures, highlighting the need for improved gesture recognition under varied conditions.

Figure 5.1

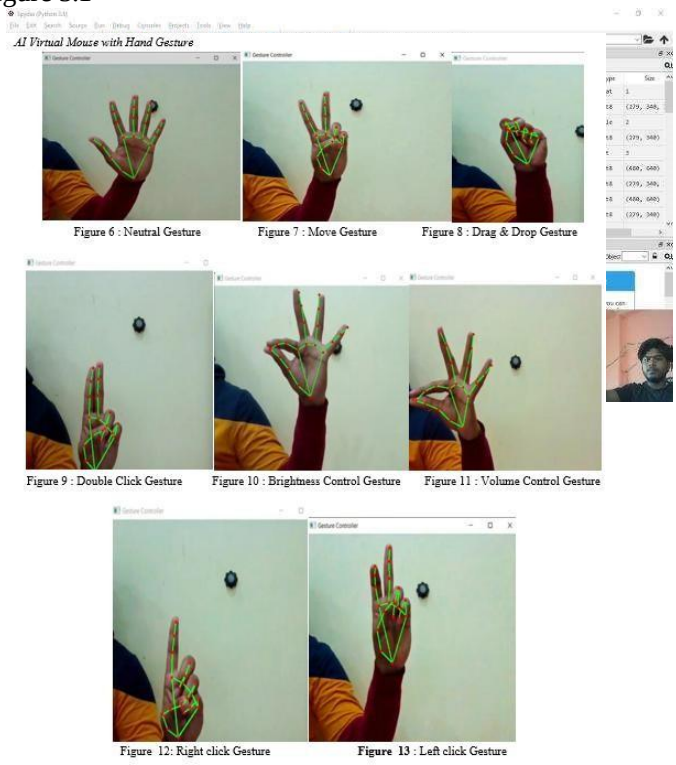

Figure 5.2

Voice command integration was similarly effective, allowing users to execute predefined actions or commands, such as opening applications or clicking on items, using simple vocal instructions. The system demonstrated good accuracy in voice recognition, but occasional delays occurred when interpreting commands in noisy environments or when multiple commands were issued in quick succession. To improve performance, the voice command system could benefit from noise filtering algorithms and more advanced natural language processing capabilities to handle complex, multi-step commands. While both hand gestures and voice commands are promising input methods for the "Optimanus Synaptic Interface," further optimization is required for them to function seamlessly in real-world scenarios. Enhancing the gesture recognition system for better precision and refining the voice command interface to handle noise and more complex instructions will significantly improve the system's overall performance.

V. Conclusion

To sum up, the combination of hand gestures and voice commands shows great promise for improving cursor control and user interaction. The testing phase showed that both methods are flexible and easy to use, offering a fresh alternative to traditional input devices. Hand gestures, in particular, allowed users to control the cursor through natural movements, making the system intuitive and fluid. Users could perform basic tasks accurately, showing that gesture control could be useful for many different applications. However, the system faced challenges when users needed to make precise or quick movements, highlighting the need for improvements in gesture recognition. The system should be fine-tuned to better detect subtle movements and work well in more dynamic environments, especially for tasks that require high precision.

Similarly, the voice command system worked well in quiet settings, allowing users to control the cursor and carry out actions like selecting or clicking by speaking simple commands. However, it struggled in noisy environments and when users gave multiple commands too quickly. This indicates that while voice commands are useful, they need to be more reliable in real-world situations. Adding noise reduction features and improving the system's ability to understand complex or multi-step commands could make the voice command system more effective in everyday use.

One of the strengths of combining these two methods is the ability for users to switch between hand gestures and voice commands easily or even use both at the same time. This provides a flexible and efficient way to interact with the system. However, problems arose when the system had to process both types of inputs at once, such as when a user spoke a command while making a

gesture. This sometimes led to delays or errors. To fix this, the

system needs better algorithms to handle conflicting inputs and prioritize them effectively. Using AI to decide which input method to prioritize based on the context could greatly improve the system's performance.Overall, while both hand gestures and voice commands show strong potential, there is still work to be done. Improvements in gesture recognition, noise handling for voice commands, and more seamless integration of the two methods are needed. By addressing these issues, the system could become even more adaptive and user-friendly, with wide applications in areas like assistive technology and more efficient human- computer interaction across various fields.

Refrences

[1] "Real-time virtual mouse system using RGB-D images and fingertip detection" Dinh-Son Tran1 & Ngoc-Huynh Ho1 & Hyung-Jeong Yang1 & Soo-Hyung Kim1 & Guee Sang Lee ,Published Date23/11/2020

[2] Masurovsky, A., Chojecki, P., Runde, D., Lafci, M., Przewozny, D., Gaebler, M.,2020. Controller-Free Hand Tracking for Grab-and Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study.MultimodalTechnol.Interact.4,91. https://doi.org/10.3390/mti4040091.

[3] J. Katona, "A review of human–computer interaction and virtual reality research fields in cognitive Info Communications," Applied Sciences, vol. 11, no. 6, p. 2646, 2021.

[4] B. J. Boruah, A. K. Talukdar and K. K. Sarma, "Development of a Learning-aid tool using Hand Gesture Based Human Computer Interaction System," 2021 Advanced Communication Technologies and Signal Processing (ACTS), 2021, pp. 1-5, doi: 10.1109/ACTS53447.2021.9708354.

[5] Y. Zhou, G. Jiang & Y. Lin, "A novel finger and hand pose estimation technique for real-time hand gesture recognition," in Pattern Recognition vol. 49, pp. 102-114, January 2016.

[6] K. S. Varun, I. Puneeth and T. P. Jacob, "Virtual Mouse Implementation using Open CV," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 435-438, doi: 10.1109/ICOEI.2019.8862764

[7] Vijay Kumar Sharma, Vimal Kumar, Sachin Tawara and Vishal Jayaswal, "Virtual Mouse Control Using Hand Class Gesture", *Gis Science Journal*, vol. 7, 2020.

[8] S. R. Chowdhury, S. Pathak and M. D. A. Praveena, "Gesture Recognition Based Virtual Mouse & Keyboard", *2020 4th International Conference on Trends in Electronics & Informatics (ICOEI)*, no. 48184, pp. 585-589, 2020.

[9] M. Ranawat, M. Rajadhyaksha, N. Lakhani and R. Shankarmani, "Hand Gesture Recognition Based Virtual Mouse Events", *2 nd International Conference for Emerging Technology (INCET)* , 2021.

[10] K. H. Shibly, S. Kumar Dey, M. A. Islam and S. Iftekhar Showrav, "Design & Development of Hand Gesture Based Virtual Mouse", *2019 1 st International Conference on Advances in Science Engineering & Robotics Technology (ICASERT)*, pp. 1-5, 2019.

[11] Aviral Gupta and Neeta Sharma, "A REAL TIME CONTROLLING COMPUTER THROUGH COLOR VISION BASED TOUCHLESS MOUSE", *Parishodh Journal*, vol. IX, no. III, pp. 5077, March 2020, ISSN ISSN NO: 2347-6648.

[12] N Steven Raj, Veeresh S Gobbur, Rahul Patil Praveen and Veerendra Naik, "Implementing Hand Gesture Mouse Using OpenCV", *International Research Journal of Engineering & Technology (IRJET)*, vol. 07, no. 06, June 2020, ISSN e-ISSN: 2395-0056.

[13] V. V. Reddy, T. Dhyanchand, G. V. Krishna and S. Maheshwaram, "Virtual Mouse Control Using Colored Fingertips & Hand Gesture Recognition", *2020 IEEE-HYDCON*, pp. 1-5, 2020.

[14] K. H. Shibly, "Design and Development of Hand Gesture-Based Virtual Mouse", *In Proceedings of the IEEE Conference on ICASERT*, 2019, [online] Available: https://ieeexplore.ieee.org/document/8934612).

[15] Hritik Joshi, Nithin waybhase, L Ratnesh and M Dharmendra, "Design of virtual mouse using gesture recognition and machine learning", *Research square 2022*, 2022, [online] Available: https://assets.researchsquare.com/files/rs-1616375/v2/9df1e90e-660f-4fdb-83a4-446bc6e738f9.pdf?c=1654102964.

[16] P. Rajendra, P. Kaushik, A. Gautam and S. Saha, "AI-Based Virtual Mouse Control Using Deep Learning Techniques", *IEEE 2nd International Conference on Sustainable Energy Electronics and Computing Systems (SECS)*, pp. 647-652, 2020

[17] N. Mohamed, M. B. Mustafa and N. Jomhari, "A Review of the Hand Gesture Recognition System: Current Progress and Future Directions," in IEEE Access, vol. 9, pp. 157422-157436, 2021.

[18] Oudah, Munir, Ali Al-Naji, and Javaan Chahl. 2020. "Hand Gesture Recognition Based on Computer Vision: A Review of Techniques" Journal of Imaging 6, no. 8: 73.

[19] Devanshu Singh, "Virtual Mouse using OpenCV", *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 12, pp. 1055-1058, 2021.

[20] G.R.S. Murthy and R. S. Jadon, "Hand gesture recognition using neural networks", *Advance Computing Conference (IACC)*, vol. 138, pp. 134, Feb. 2010.

[21] Akshay Ishwar Pawar and Sahil Prakash Tumbare, Mouse Control using a Web Camera and Hand Gestures with Colour Tapes, 2016.

[22] Banda Aneela, "Implementing a Real Time Virtual Mouse System and Fingertip Detection based on Artificial Intelligence", *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. VI, pp. 2265-2270, 2021.

.