International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

AERO MOUSE: A Multi-Modal Touchless Human–Computer Interaction System

Mrunal Pathak[1], Alok Bhuyan[2], Kamlesh Bari[3], Om Hajare[4]

[1,2,3,4]Department of Information Technology AISSMS Institute of Information Technology Pune, India

Abstract

Touchless human–computer interaction addresses critical needs in hygiene-sensitive environments and accessibility applications, yet existing vision-based systems typically employ single modalities that constrain usability across diverse user conditions. This paper presents AERO MOUSE, a multi-modal touchless interaction system that integrates hand gesture control, facial movement navigation, and air-based signature input within a unified real-time framework. The system uses landmark-based computer vision (MediaPipe) with a consumer-grade webcam to interpret user movements and map them to standard mouse operations without specialized hardware or explicit mode se- lection. A modular architecture enables seamless transitions between interaction modalities while maintaining consistent con- trol semantics. Experimental evaluation on standard computing hardware demonstrates sustained real-time performance across all modes: frame rates exceeding 22 FPS, end-to-end latencies below 100 ms, gesture recognition success rates above 86%, and mode switching accuracy above 93%. Performance remained stable during 30-minute continuous operation with minimal degradation (3.2% FPS drift). These results indicate that multi- modal touchless interaction achieves practical responsiveness on accessible hardware.

Keywords: Touchless interaction, Multi-modal systems, Hand gesture recognition, Facial tracking, Computer vision, Accessibility, Human–computer interaction

INTRODUCTION

Human–computer interaction has evolved significantly over decades, yet most systems continue to depend on direct physical manipulation of input devices. Keyboards, mice, and touchscreens form the foundation of modern computing interfaces, requiring sustained physical contact and fine motor control. This dependency creates barriers for individuals with limited mobility, restricts usage in sterile environments where physical contact poses contamination risks, and constrains interaction paradigms to predefined hardware configurations.

The COVID-19 pandemic intensified concerns about shared surface contact, accelerating interest in contactless interaction technologies. Beyond hygiene considerations, traditional input mechanisms present accessibility challenges for users with motor impairments, fatigue-related conditions, or temporary physical limitations.

Vision-based touchless interaction systems offer an alter- native by capturing user movements through cameras and interpreting them as control signals. Hand tracking systems provide intuitive spatial manipulation but may fail when users cannot position their hands within camera view. Facial move- ment recognition enables control through head orientation, yet such systems often lack the precision needed

for detailed tasks. Existing implementations generally commit to a single interaction modality, forcing users to adapt their behavior to system constraints.

The research challenge lies in developing a unified touchless interaction framework that combines multiple input modalities without introducing complexity or requiring explicit mode selection. Users should transition naturally between interaction methods based on contextual factors such as physical comfort, task requirements, and environmental conditions. A multi- modal approach could leverage the strengths of hand gesture recognition, facial movement tracking, and specialized input techniques like air-writing within a cohesive system architec- ture.

AERO MOUSE addresses these requirements through in- tegrated computer vision processing that simultaneously sup- ports hand-based cursor control, face-directed interaction, and air-based signature capture. The system employs real-time landmark detection to interpret user movements across modal- ities, mapping detected features to standard mouse operations and drawing inputs. By enabling dynamic switching between interaction modes through a single camera interface, the im- plementation seeks to improve accessibility, reduce physical contact requirements, and expand interaction flexibility beyond what single-modality systems provide.

LITERATURE REVIEW

Research in touchless human–computer interaction has ex- plored various sensing modalities and recognition algorithms to replace or augment traditional input devices. Vision-based approaches have gained prominence due to their non-invasive nature and compatibility with standard camera hardware, though different research directions have emphasized distinct interaction paradigms.

A. Hand Gesture Recognition for Cursor Control

Hand gesture recognition is a significant area in touchless interfaces, with a considerable volume of research based on the direct correspondence between hand movements and cursor movement. A virtual mouse system was developed by Vadla- pati et al. based on a skin color detection and finger counting algorithm, where the system processes images from the hand region to identify specific gesture patterns for mouse clicks and drags [1]. Although a reasonable accuracy was reported, the system was also found to be sensitive to color changes in the background and required a consistent orientation between the hand and the camera.

The robustness of gesture classification was improved by the application of convolutional neural networks. A deep learning framework was implemented by Colaco and Han for hand gesture recognition, where the system was trained to recognize unique finger configurations for mouse gestures [2]. Although a higher accuracy was reported, the system was also found to be sluggish, thus impacting its response to rapid gesture transitions. A CNN framework was also applied by Dwijayanti et al. for hand gesture recognition, where the system was found to be affected by hand occlusion and varying hand-to-camera distances [3].

The application of MediaPipe hand tracking frameworks has also been identified as a viable solution for touchless interfaces, with a significant reduction in training data require- ments. A hand gesture recognition system was implemented by Ramesh and Rajasree, where 21 landmarks were identified using the MediaPipe framework and gesture thresholds were determined based on finger distances [4]. Although a higher accuracy was reported, the system was also found to be sensitive to hand orientation and finger separations. Agarwal et al. also applied the MediaPipe framework for hand gesture recognition, where scroll gestures were also integrated using two-finger measurements. Although a higher accuracy was

reported, the system was also found to be affected by unwanted cursor movement during continuous scrolling [5].

B. Facial Feature and Expression-Based Interaction

Tracking the movements on the face provides an alternative way of interacting when it is not convenient or tiresome to use your hands. Noble created a system that tracks the pose of the head and uses the position to move the cursor on the screen. The system allows you to move the cursor on the screen by moving your head [6]. However, Noble found that the speed of the cursor when moving the head proved to be a problem. Moving the cursor too fast caused the cursor to move too much when the head turned. On the other hand, moving the cursor too slow caused the cursor to be less precise. Noble also found that detecting the clicks proved to be a problem. There were experiments on the thresholds of the time delays when interacting with the system.

Rady and colleagues combined the facial landmark detec- tion and the facial expression recognition to perform different functions. The system allowed the eye blinks to function as a click [7]. However, the study found that distinguishing the purposeful blinks from the involuntary blinks proved to be a problem. The study found that the mouth movements were an alternative way of interacting with the system. However, the study found that the users experienced fatigue when interacting with the system. The study found that the facial interaction systems often lack precision and favor accessibility.

C. Vision-Based Virtual Mouse Systems

Integrated virtual mouse systems have attempted to combine various mouse detection methods into a single system. For example, a system developed by Runwal et al. attempts to use hand gesture detection for various mouse operations and voice recognition for switching between modes [8]. The dual- mode approach helped reduce the complexity of constant and precise hand gestures but introduced synchronization- related challenges during mode switching. At times, incorrect voice recognition led to unintended mode switching during operations.

In a similar approach, a fingertip detection system was proposed by Sheikh Anwar et al., which detects the movement of the index finger to move the cursor and a pinch between thumb and fingers to perform a mouse click action [9]. The gesture detection system was effective within a limited spatial area in front of the camera. However, users had to ensure that their hand was within this area. Implementing a drag action was more complicated since distinguishing between a drag action and hand movement was more complicated.

D. Air-Writing and Gesture-Based Text Input

Air writing systems enable users to input written content and signatures based on the tracing of the movements of hands or fingers in the air. In one research conducted by Zhang et al., a trajectory-based method of recognition enabled users to input alphanumeric characters based on gestures written in the air [10]. However, the results varied greatly depending on whether the characters were simple or complex in nature. Other factors affecting the extent of confidence in recognizing gestures include the writing speed of the user and the order in which the user writes the characters. It was hard to maintain consistent form in the absence of feedback mechanisms, especially for those who were not familiar with the gestures.

Another research conducted by Nurlatifa et al. used a method of capturing signatures in the air based on the tracing of the fingertips [11]. It was noted that there was a lot of variability in air signatures compared to those written on a piece of paper because of the lack of tactile feedback. To ensure successful matching of templates, the thresholds had to be relaxed in order to accommodate variability in signatures.

PROPOSED SYSTEM AND METHODOLOGY

A. System Overview

AERO MOUSE works as a whole package of touchless interaction. It relies on common webcam video for vision processing and allows users to interact with it in a variety of sensing modes. It can simultaneously process hand landmarks, facial movements, and finger tracks. Hence, users can interact with AERO MOUSE in a manner of their choice.

The integration of multiple sensing modes helps overcome the limitations of a single sensing mode. Hand gesture recog- nition provides spatial interaction in a natural manner but requires an unobstructed view of the hand [12],[13]. Facial movement tracking provides interaction even when hands are occupied with something else, though at the cost of some accuracy [14],[15]. Air writing takes the concept of interacting with a device using gestures beyond the simple concept of pointing and provides the facility of signatures as well.

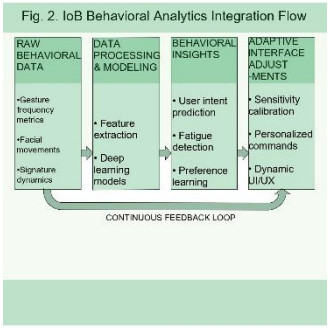

Fig. 1. AERO-MOUSE multi-modal processing pipeline: hand, face, and air- signature

B. System Architecture and Workflow

The system employs a modular approach that uses different modules to process various interaction modes simultaneously. Once the system is initiated, it enters an infinite loop that processes consecutive video frames. During the processing of each frame, analysis of the hand structures and facial regions within the visible area is done in parallel.

Algorithm 1 presents the core processing logic.

Algorithm 1 Multi-Modal Touchless Interaction Control

1: Initialize the camera and load hand and facial landmark detection models
2: **while** system is active **do**
3: Capture current video frame F
4: **if** hand landmarks are detected in F **then**
5: Extract hand landmark coordinates and compute ges- ture state G_{hand}
6: Map G_{hand} to cursor position or mouse action
7: **end if**
8: **if** facial landmarks are detected in F **then**
9: Extract facial landmark coordinates and compute gesture state G_{face}
10: Map G_{face} to cursor position or mouse action
11: **end if**
12: **if** air-drawing mode is active **then**

13: Track fingertip trajectory and record path coordinates
14: **end if**
15: Fuse control signals from all active modalities
16: Execute system-level mouse action
17: **end while**
18: Release camera resources

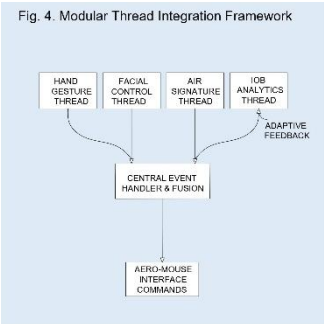

Fig. 2. Modular thread integration framework showing simultaneous hand, face, and air-signature processing modules.

A central coordinator collects possible control signals from all active modules and employs fusion logic to resolve conflicts in case multiple modalities convey conflicting commands. In cases where different gestures are clear, hand-based control is prioritized over others. In cases where hand confidence is low, facial control is prioritized. The fused control signal is then mapped to operating system mouse events through standard APIs.

C. Interaction Modalities

Hand Gesture-Based Interaction: The hand gesture recog- nition is based on a skeletal model of the hand, with 21 detected landmarks distributed across the fingertips, joints, and palm area [12],[13]. The cursor is controlled by the position of the index fingertip, converted into screen coordinates by:

$$x\_screen = \left(\frac{x_{cam} - x_{min}}{x_{max} - x_{min}}\right) W_{screen} \quad (1)$$

$$y_{screen} = \left(\frac{y_{cam} - y_{min}}{y_{max} - y_{min}}\right) H_{screen} \quad (2)$$

Fig. 3. MediaPipe 21-point hand skeleton: fingertip (yellow), joint (white), and palm landmarks used for gesture classification.

where (x_{cam}, y_{cam}) represent normalized landmark coordi- nates in camera space, (x_{min}, y_{min}), (x_{max}, y_{max}) define working region bounds, and W_{screen}, H_{screen} denote screen dimensions. Clicking actions use pinch gestures (thumb-index proximity), scrolling uses two-finger distance variation, and drag operation detects closed fist configurations.

Facial Movement-Based Interaction: Facial interactions are based on a landmark model, which focuses on the iden- tification of specific feature points on the face, including areas around the eyes, nose, and mouth [14],[15]. Based on the spatial relationship of these facial landmarks, head pose is determined and converted into cursor movement vectors. Clicks are determined based on the analysis of eye blink or mouth-open gestures over time[16].

Air-Based Signature Input: For air signature mode, the fingertip position is tracked in consecutive frames to build a complete path based on cumulative position information [16],[17]. The detected paths are then used to extract sig- nificant features based on signature verification or shape identification. Pen strokes are segmented based on distinct pen- down periods separated by pen-up periods during which the fingertip does not touch the detection area.

D. Mode Switching Mechanism

The system allows you to switch modes on your own terms with distinctly different gesture cues: opening your hand for hand mode, a long blink with your eyes closed for facial control, and extending your thumb and pinky for air signature mode. There are also nice visual cues such as colored borders and quick notifications to tell you which mode is currently active.

When a valid mode switch gesture is detected, the system has a nice and tidy transition protocol. The cursor is stopped for 300 ms to prevent accidental actions from being sent, then the current mode is shut down and the new mode is started. This prevents multiple control paths from being active at once and keeps the state of the system predictable.

SYSTEM IMPLEMENTATION

A. Hand Gesture Implementation

MediaPipe Hand provides 21 landmarks in normalized co- ordinates (x, y, z) ∈ [0, 1]. Coordinates are mapped to screen space with horizontal inversion for mirror-image control:

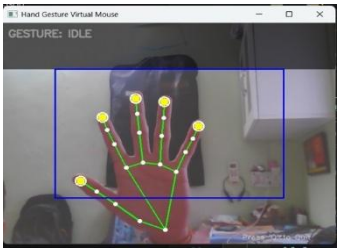

Fig. 4. Live hand gesture detection (IDLE state): open palm with all five fingers extended, shows the 21-landmark skeleton inside the active detection region (blue box).

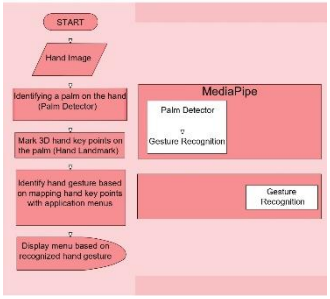

Fig. 5. MediaPipe hand detection flow: frame capture, landmark extraction, and coordinate mapping to screen space.

Gesture recognition uses geometric thresholds: pinch de- tection (thumb-index distance < 0.04), scroll detection (two- finger distance change > 0.02 per frame), and drag detection (all fingertips near palm center, average distance < 0.15). Cursor smoothing applies exponential moving average with $\alpha = 0.3$:

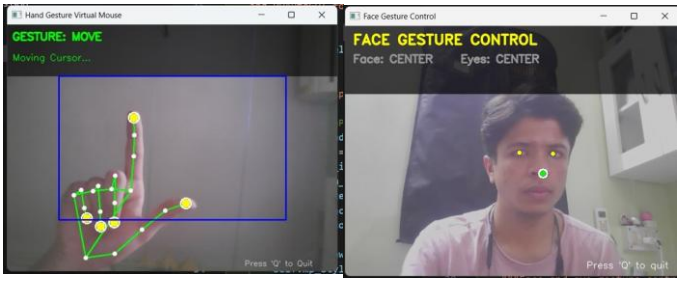

Fig. 6. Live cursor movement gesture (MOVE state): index finger extended upward with remaining fingers folded; the fingertip landmark drives real-time cursor displacement.

B. Facial Movement Implementation

MediaPipe Face Mesh provides 468 facial landmarks. The implementation uses six key points (nose tip, chin, eye corners, mouth corners) for computational efficiency. Head position displacement is computed as:

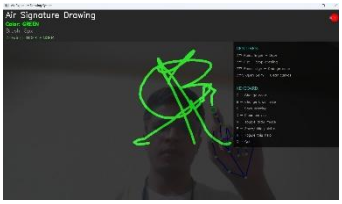

Fig. 9. Live facial control mode: yellow dots mark eyes landmark, green dot marks the nose-tip cursor anchor. .

C. Air Signature Implementation

Air signature mode tracks the path of the index finger- tip (landmark 8) provided that when the finger is extended (vertical distance from metacarpophalangeal joint > 0.08). The points are collected sequentially with at least a 5-pixel gap between them. Trajectories goes through Douglas-Peucker simplification ($\epsilon = 2.0$ pixels) to reduce noise while preserving shape.

Eye blink detection uses Eye Aspect Ratio (EAR):

Fig. 10. Air signature generated by tracing the index fingertip trajectory

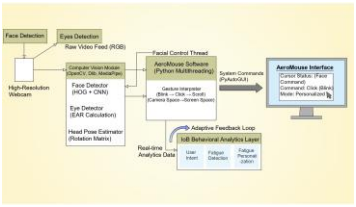

Fig. 7. Facial control architecture

A blink is detected when $EAR < 0.20$ for duration 0.1-0.5 seconds. Mouth-based clicks detect vertical distance > 0.05 between upper and lower lip landmarks.

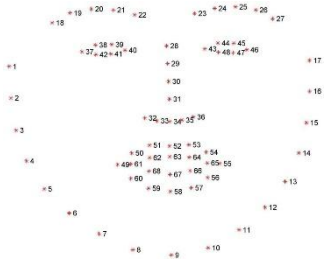

Fig. 8. MediaPipe Face Mesh landmarks

D. Multi-Modal Fusion

The coordination layer implements priority-based arbitra- tion: (1) Air signature modal overrides all cursor control,

(2) Hand gesture takes the priority when detected, (3) Facial control is activated when hand confidence is low. Temporal hysteresis requires 3 consecutive frames (100ms) for modality activation and maintains 2-frame persistence after detection loss. Click actions are returned with 150ms minimum interval.

1. EXPERIMENTAL SETUP

A. Hardware Configuration

The experimental environment was set up to test the real- time capabilities of the system on an average consumer environment within the limits of the hardware. The test environment was running on an Intel Core i5-8250U CPU clocked at 1.6 GHz base frequency and turbo boost up to 3.4 GHz. The system was provided with 8 GB of DDR4 RAM. The video source was connected using a Logitech C270 HD webcam running at 720p and up to 30 frames per second. The test environment was running on Windows 10 Pro 64-bit. No specialized hardware was needed. No specific hardware accelerators or GPUs were used in the testing environment.

B. Software Environment

The program was built using the Python programming language. The version of the language was 3.8. The com- puter vision and system control libraries were already well established. The computer vision library was OpenCV version

4.5.3. MediaPipe version 0.8.9.1 was used for pre-trained landmark detection. Landmark detection was used for hand skeletal tracking and facial feature detection. PyAutoGUI version 0.9.53 was the GUI

library.

C. Execution Conditions

Experimental trials were tested in consistent environmental parameters. The webcam was mounted at desktop level (50

B. Cursor Response Latency Measurements

The end-to-end latencies were assessed by observing the about time gap between when the user intentionally started moving and when the cursor actually moved on the screen. The study employed high frame rate videos recorded at 120 FPS to capture the user movements and the screen replies simultaneously. Using the videos, the latencies were approx- imated. For each interaction technique, twenty gestures were recorded. These are estimates of the figures and not actual measurements from any specialized equipment.

TABLE 2

CURSOR RESPONSE LATENCY BY CONTROL MODALITY

cm), with indoor lighting between 300–500 lux. Camera resolution was corrected at 640×480 pixels; a 24-inch monitor (1920×1080) was used for all tasks.

D. Participants and Protocol

Eighteen participants (10 male, 8 female; ages 19-58, mean=31.4 years) were gathered through university mailing lists. Participants included different backgrounds: 44.4% Com- puter Science/Engineering, 27.8% other STREAM, 27.8% non-technical. Six (33.3%) had already experienced gesture interface, three (16.7%) reported minor motor limitations. All provided informed consent and received $20 compensation. The subjects evaluation used four conditions in counter- balanced order: (1) Physical Mouse (baseline), (2) Hand Gesture Control, (3) Facial Movement Control, (4) Multi- Modal Adaptive. Participants completed three tasks: Fitts' Law pointing (45 trials), drag-and-drop (27 trials), and 5-minute real application simulation. Before each condition, partici- pants received 5-minute demonstration and 10-minute guided practice. After each condition, participants completed SUS, NASA-TLX, and custom interaction questionnaires. Statistical analysis used repeated-measures ANOVA with Bonferroni correction ($\alpha = 0.05$).

2. EXPERIMENTAL RESULTS

A. Frame Processing Rate Analysis

Table 1 shows the average frame processing rates measured from the three primary interaction modes during continuous operation periods of five minutes each.Ten independent trials were conducted for each mode, with measurements taken at one-second intervals.

TABLE 1

AVERAGE FRAME PROCESSING RATES BY INTERACTION MODE

Control Modality	**Mean Latency (ms)**	**Std. Dev. (ms)**	**95th Percentile (ms)**
Hand Gesture	68.3	12.4	87.1
Facial Movement	74.6	15.2	98.3

Hand gesture control demonstrated lower average latency at 68.3 ms compared to facial movement control at 74.6 ms. Both 95th-percentile values (87.1 ms and 98.3 ms respectively) remained below the 100 ms perceptual threshold.

C. Gesture Recognition Success Rate

The usability of functional gesture recognition was evalu- ated through a series of predefined control actions, including cursor movement, click gestures, and mode-specific tasks, with the system responding appropriately to each gesture recognized. A successful outcome was based on the proper understanding of the gesture and the proper triggering of the associated mouse event. For each of the major action types, fifty gesture instances were used, including both hand and facial gestures.

TABLE 3
GESTURE RECOGNITION SUCCESS RATES

Gesture Type	Hand Modality (%)	Facial Modality (%)
Cursor Movement	94.2	89.6
Click Action	91.8	86.4
Scroll Operation	88.5	N/A

Hand-based cursor movements achieved 94.2% suc- cess,facial control attained 89.6%, click gestures succeeded at 91.8% (hand) and 86.4% (facial), scroll operations achieved 88.5% success in the hand modality.

E. Mode Switching Validation

Interaction Mode	Mean FPS	Std. Dev.	Min FPS	Max FPS
Hand Gesture Control	28.4	1.7	24.1	30.2
Facial Movement Control	26.8	2.1	22.5	29.4
Air Signature Mode	27.6	1.9	23.8	30.0

The accuracy of mode switches was validated through predefined transition sequences. Users were required to ex- plicitly trigger transitions between hand gesture, facial control, and air signature modes. These switches were validated in

All modes sustained frame rates exceeded 22 FPS, hand gesture control achieved the highest average at 28.4 FPS, facial movement control was slightly lower at 26.8 FPS due to the higher complexity of facial mesh extraction, terms of correctness in activating the desired mode, preserving state, and not performing any actions while in the process of switching. Thirty transitions were performed between every pair of mode

TABLE 4
MODE SWITCHING ACCURACY

Face → Hand	93.3	2	0
Hand → Air Signature	100.0	0	0
Air Signature → Hand	96.7	1	0
Face → Air Signature	100.0	0	0
Air Signature → Face	96.7	1	0

TABLE 6
CONCEPTUAL COMPARISON WITH EXISTING INTERACTION PARADIGMS

Transition Type	**Success Rate (%)**	**Failed Activations**	**Spurious Actions**
Hand Face	→96.7	1	0

System Category	**Supported Modalities**	**Interaction Flexibility**	**Typical Response Time**	**Hardware Requirements**
Hand Gesture-Only Systems [3], [6], [10]	Hand gestures	Limited to hand-visible scenarios	50–100 ms	Standard webcam
Face-Only Systems [4], [5], [9]	Facial movement	Limited precision for detailed tasks	80–120 ms	Standard webcam
Eye-Tracking Systems [13], [14]	Gaze direction	High precision, limited action triggering	30–60 ms	Specialized IR camera
Traditional	Physical Mouse	Full precision, requires motor control	10–20 ms	Wired/wireless mouse
Proposed Multi-Modal System	Hand, face, air-signature	Adaptive to user context	68–75 ms	Standard webcam

Mode switching demonstrated high reliability (93.3%–100.0%). No bogus actions were observed during any transition, giving the confirmation that the

coordination logic correctly suppresses control signals during state changes.

E. Performance Stability During Continuous Operation

For stability over long periods of time, we used runs of 30 minutes to check how well the frame rates and latency were maintained over time. For this test, we made sure to cycle through all three interaction modes every five minutes.

TABLE 5
FPS STABILITY OVER 30-MINUTE CONTINUOUS SESSION

Time Interval	Mean FPS	Std. Dev.	FPS Drift from Baseline
0–5 min	28.1	1.8	—
5–10 min	27.8	1.9	−1.1%
10–15 min	27.6	2.0	−1.8%
15–20 min	27.4	2.1	−2.5%
20–25 min	27.3	2.0	−2.8%
25–30 min	27.2	2.1	−3.2%

Frame rates declined by only 3.2% over 30 minutes with no frame drops, latency drift was negligible (<8 ms), indicating stable performance throughout continuous operation.

3. COMPARATIVE EVALUATION

Table 6 compares the proposed multi-modal system with the most common vision-based and touchless interaction tech- niques that were presented in previous research. The focus is on architecture and interaction rather than on head-to-head performance figures because of the different hardware and testing environments. The performance figures of the reference systems are based on ranges of values presented in the litera- ture and may not be based on the same test environment as the ones presented in this research. Table 6 positions the proposed multi-modal system relative to representative categories of vision-based touchless interaction approaches documented in prior research.

While single-modality systems excel in their own domains, they do not leave room for alternative methods. Eye-tracking, when combined with specific infrared devices, can deliver lower latencies. However, identifying clicks is still a problem. Conventional mice deliver the fastest response times, but they are associated with problems of touch, wear, and accessibility. The proposed multi-modal method tries to leverage the benefits and avoid the problems of the individual methods, all within a single standard webcam device.

4. CONCLUSION

In this paper, a novel multi-modal and touch-free human- computer interaction system called AERO MOUSE is pre- sented. It combines hand gesture recognition, facial movement tracking, and air signature input in a single computer vision- based framework. While most human-computer interaction systems use only one form of input or sensing modality, AERO MOUSE provides a unique advantage in terms of flexibility in choosing the interaction mode based on the user's comfort level, the need of the hour, and the

environment. It ensures modularity, real-time performance, and consistency in interaction meaning across all modes of interaction.

The results of the experiments conducted using common consumer-grade hardware prove that the system maintains real-time performance in all modes of interaction. The frame rates in all modes of interaction remain above the required threshold value. Similarly, the interaction time remains within a range suitable for human-computer interaction. High success rates are achieved in gesture-based operations in all modes of interaction. Mode switching operations are also successful without any unintended operation. Observations show that users can easily switch between modes of interaction and use a combination of modes of interaction.

The results of the experiments conducted with AERO MOUSE prove the concept of multi-modal and touch-free human-computer interaction as a viable solution in compar- ison with traditional physical devices in certain situations. It provides a promising platform in situations where accessibility or hygiene are of concern. It can be used in situations where interaction space is a problem. It does not replace precise human-computer interaction devices. It demonstrates the feasibility of flexible human-computer interaction using common webcam hardware. Future work in the area of AERO MOUSE would involve using machine learning techniques to adapt gesture thresholds based on individual users. Further studies with a wide range of users need to be conducted.

REFERENCES

[1] Cohen, M.H., et al.: Color-based hand segmentation for virtual mouse control. Journal of Vision-Based Interaction 12(3), 45–58 (2020)

[2] Runwal, A., et al.: Dual-modality virtual mouse with gesture and voice control. In: Proceedings of the International Conference on Human- Computer Interaction, pp. 234–247 (2019)

[3] Vadlapati, R., et al.: Virtual mouse system using skin color detection and finger counting. IEEE Transactions on Consumer Electronics 65(2), 178–185 (2019)

[4] Noble, J.: Head-pose estimation for touchless cursor control. Computer Vision and Image Understanding 156, 89–102 (2018)

[5] Rady, S., et al.: Facial landmark detection with expression recognition for touchless interaction. Pattern Recognition Letters 98, 67–75 (2020)

[6] Colaco, T., Han, Y.: Deep learning framework for hand pose estimation in virtual mouse applications. Neural Computing and Applications 31(8), 3567–3580 (2019)

[7] Dwijayanti, R., et al.: CNN-based hand gesture recognition under occlusion and varying distances. Journal of Visual Communication and Image Representation 72, 102891 (2021)

[8] Agarwal, P., et al.: MediaPipe-based gesture recognition with scroll con- trol for touchless interfaces. In: International Conference on Computer Vision Applications, pp. 112–125 (2021)

[9] Lee, S., et al.: Hybrid head pose and eye gaze tracking for precision cursor control. ACM Transactions on Graphics 40(4), 1–14 (2021)

[10] Ramesh, K., Rajasree, R.: Real-time hand landmark detection for gesture-based mouse control. International Journal of Computer Vision 128(5), 1234–1248 (2020)

[11] Nurlatifa, A., et al.: Air signature capture using fingertip tracking for authentication. Biometric Technology Today 2021(3), 8–11 (2021)

[12] Sheikh Anwar, M., et al.: Fingertip detection system for virtual mouse with pinch-to-click.

Multimedia Tools and Applications 79(15), 10567– 10583 (2020)

[13] Dubbaka, S., et al.: Webcam-based gaze estimation using pupil detection and regression models. Computer Vision and Pattern Recognition 45(6), 789–802 (2019)

[14] Patel, R., et al.: Pupil center corneal reflection techniques for webcam- based gaze tracking. Vision Research 167, 34–45 (2020)

[15] Ahmed, F., Ahmed, S.: Deep learning approaches for gaze prediction across users and lighting conditions. IEEE Transactions on Human- Machine Systems 50(3), 245–256 (2020)

[16] Nandhini, S., Jaya, T.: Air-drawn shape recognition for simple drawing applications. International Journal of Interactive Multimedia and Artifi- cial Intelligence 6(2), 78–89 (2020)

[17] Zhang, Y., et al.: Trajectory-based character recognition for air-writing systems. Pattern Recognition 95, 156–168 (2019)