Low-Cost Gesture Guided Swarm Control Using MediaPipe and Decentralized Behavior

Ishan Zadbuke[1], Sahil Panchavishe[2], Vivek Khandelwal[3], Yashwardhan Abhale[4] and Anshul Jain[5*]

[1,2,3,4,5] COEP Technological University, Pune, Maharashtra, India

*Corresponding author: anshul_ajmera2007@yahoo.co.in

Abstract. Human–Swarm Interaction (HSI) supports straightforward control of groups of robots, but many current systems depend on costly sensing equipment like motion capture rigs, depth cameras, or wearable devices, which limits their use in educational and resource constrained environments. In this work, we propose a low-cost, vision based HSI system that uses a standard monocular webcam and the MediaPipe library to help gesture driven control of a ground robot swarm. The system extracts 21 hand landmarks and converts them into a normalized 42 dimensional feature representation for real time gesture recognition using simple geometric rules. High level commands are broadcast to the swarm using a centralized star network topology with User Datagram Protocol (UDP) broadcasting, while execution remains decentralized at the robot level. Robustness to packet loss and network jitter is achieved through a latching based control mechanism. Experimental results show an average gesture recognition accuracy of 95.0% with a vision processing latency of approximately 32 ms per frame, and an estimated end to end system latency of about 87 ms, confirming suitability for real time interaction.

Keywords: Human Swarm Interaction, Swarm Robotics, Computer Vision, MediaPipe, Decentralized Control.

1 Introduction

Swarm robotics draws inspiration from collective behaviours observed in nature, such as ant colonies, bee hives, and bird flocks, where complex group intelligence emerges from simple local interactions among individuals [1], [2]. The implementation of decentralized control systems helps robot swarms to attain their maximum operational capacity while executing tasks such as environmental monitoring, search and rescue, and precision agriculture. Within this paradigm, Human–Swarm Interaction (HSI) plays an important role by enabling a human operator to guide collective behaviour at the swarm level, translating high level intent into coordinated robot actions.

Despite significant progress, the practical adoption of HSI remains constrained by the cost, complexity, and infrastructural requirements of existing interfaces. Many gesture-based swarm control systems rely on motion capture setups, depth sensors, or wearable devices. For example, Alonso Mora et al. employed gesture-based interaction using external motion capture systems, while Suresh and Martinez utilized wearable armband sensors for formation control [3], [4]. Although these

S. Sharma et al. (eds.), *Proceedings of the International Conference on Recent Advances in Intelligent and Sustainable Technologies (RAIST 2026)*, Atlantis Highlights in Intelligent Systems 17,
https://doi.org/10.2991/978-94-6239-707-1_19

approaches provide high precision, they create strict limits on environmental conditions which make it harder for people to access educational facilities and research environments that lack resources.

Recent advances in computer vision have enabled marker-less hand tracking using monocular cameras; however, important gaps persist. Several vision based HSI systems depend on depth cameras (e.g., Kinect or LiDAR), focus primarily on aerial robot platforms, or are evaluated only in simulation environments [5], [6]. Additionally, many prior works emphasize gesture recognition accuracy while giving limited attention to communication topology, network latency, and end to end system response time, which are critical factors for safety and scalability in real world swarm deployments.

This work addresses these limitations by presenting a low-cost, marker-less, real time HSI framework that enables gesture driven control of a ground robot swarm using only a standard monocular webcam. The proposed system employs MediaPipe based hand landmark extraction to obtain 21 hand landmarks, which are transformed into a normalized 42 dimensional feature vector for deterministic, geometry based gesture classification. High level commands are then broadcast to the swarm for decentralized execution, avoiding centralized path planning and ensuring scalability with increasing swarm size.

Beyond perception, this work explicitly integrates communication and timing considerations into the system design. A centralized star network topology combined with User Datagram Protocol **(**UDP) broadcast communication is used to ensure simultaneous command delivery to all agents with minimal transmission overhead. To enhance robustness against packet loss and network jitter, each robot employs a latching state control strategy that maintains stable motion in the absence of transient updates. On top of this, an analytical end to end response time model is introduced, accounting for vision processing latency, wireless transmission delay, and onboard actuation time. This analysis shows that the total system response remains well within real time safety thresholds for HSI.

2 Methodology

The proposed system uses a computationally efficient, vision based pipeline to interpret human hand gestures into digital commands. The core of this methodology is the extraction of high fidelity 3D hand landmarks using the MediaPipe framework.

2.1 System Workflow

The overall processing pipeline is designed to be modular and linear. The workflow begins with the Image Acquisition module, which captures frames from a standard webcam. These frames are passed to the Feature Extraction unit (MediaPipe), which outputs a set of 21 landmark coordinates. These coordinates are then normalized and flattened into a feature vector. Finally, the Geometric Classifier evaluates the vector against predefined rules to broadcast specific commands (e.g., STOP, FORM LINE) to the robot swarm. The System workflow diagram illustrating the data flow from image capture to command broadcast is shown in Fig.1.

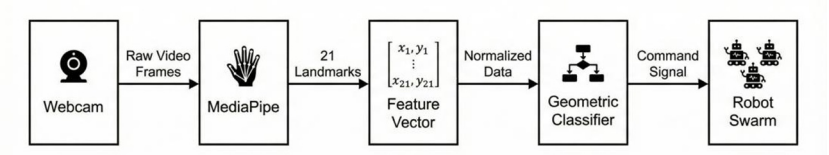

Fig. 1. System workflow diagram

2.2 Hand Landmark Extraction

The system captures video input from a standard monocular webcam. Each frame is processed using the MediaPipe Hands library [6], [8], which employs a machine learning pipeline to infer 21 3D landmarks of a hand from a single image. These landmarks represent the critical joints and fingertips, providing a skeletal model of the hand in real time. As shown in Fig. 2, the hand is represented using MediaPipe's 21 landmark points, including the wrist (0), fingertip indices (4, 8, 12, 16, 20), and intermediate joints (e.g., PIP). These landmarks provide the geometric basis for estimating finger states and extracting features for gesture classification.

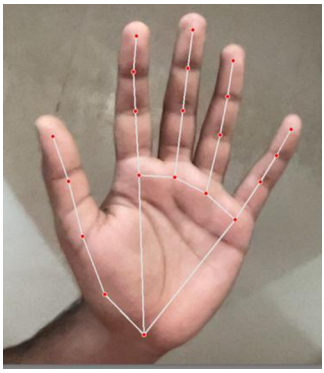

Fig. 2. MediaPipe 21 point hand landmark model used for tracking wrist, fingertips, and finger joints

3 Gesture Recognition Framework

To enable robust gesture classification without the computational overhead of deep learning models, a mathematical framework is introduced that converts raw hand landmark data into deterministic control signals.

3.1 Mathematical Hand Representation

To classify gestures robustly, the raw landmark data is converted into a structured mathematical feature vector. Let $\boldsymbol{L}$ be the set of normalized landmarks, where $\boldsymbol{i}$ represents the landmark index (i=0, 1, 2, . . ., 20). Each landmark consists of normalized spatial coordinates, as shown in Eq. (1):

$$L_i = (x_i, y_i) \tag{1}$$

To create a unified feature set for classification, these landmarks are concatenated into a fixed dimensional state vector, defined in Eq. (2):

$$H = [x_0, y_0, x_1, y_1, \dots, x_{20}, y_{20}]^T \tag{2}$$

This vector $\boldsymbol{H}$ provides the raw geometric data required to determine the state of the hand, independent of lighting conditions or skin tone.

3.2 Geometric Gesture Classification

Instead of relying on opaque neural networks, a transparent and deterministic algorithm is implemented to classify the hand state [1], [7].The system evaluates the "Open" or "Closed" state of each finger by comparing the vertical position of the fingertip ($\mathrm{y_{tip}}$) relative to its corresponding Partial Interphalangeal (PIP) joint ($\mathrm{y_{pip}}$).

In the image coordinate system (where $\mathrm{y} = 0$ is the top of the frame), a finger is considered open if the tip is physically above the knuckle, which corresponds to a lower coordinate value, as expressed in Eq. (3):

$$y_{tip} < y_{pip} \tag{3}$$

Let the binary finger state vector be defined as in Eq. (4):

$$\boldsymbol{F} = [f_T, f_I, f_M, f_R, f_P] \in \{0,1\}^5 \tag{4}$$

where each element represents the state of the thumb, index, middle, ring, and pinky fingers respectively, and

$$f_k \in \{0,1\}, f_k = 1 \text{ if finger } k \text{ is open}, f_k = 0 \text{ if finger } k \text{ is closed.}$$

The total number of extended fingers is calculated using Eq. (5):

$$T = f_T + f_I + f_M + f_R + f_P \tag{5}$$

Based on the Boolean state of the five fingers derived from the vector; the system maps the hand configuration to specific high level swarm commands:

- **POINT (Index Open):** Detected when only the index finger is extended. This maps to the "Form Line" command, instructing the swarm to align linearly.
- **VICTORY (Index + Middle Open):** Detected when both index and middle fingers are extended. This maps to the "Split Group" command, dividing the swarm into subgroups.
- **PALM (All Open):** Detected when all five fingers are extended. This maps to the "STOP" command, serving as an emergency halt.
- **FIST (All Closed):** Detected when no fingers are extended. This maps to the "WAIT" command.

To formalize the control logic, the gesture classification function ***G(F)*** is defined**,** where ***F*** represents the binary state vector of the five fingers (Thumb, Index, Middle, Ring, Pinky) and ***T*** denotes the total count of extended fingers (**T**=Σ***F***). The specific mappings from finger states to swarm commands are defined in Eq. (6) as follows:

$$G(F) = \begin{cases} WAIT\ if\ T\ = 0 \\ FORM\ LINE\ if\ F\ \in \{[0,1,0,0,0],[1,1,0,0,0]\} \\ SPLIT\ GROUP\ if\ F\ \in \{[0,1,1,0,0],[1,1,1,0,0]\} \\ STOP\ if\ T\ = 5 \end{cases} \quad (6)$$

Due to this rule-based approach, the computational cost remains low and the system can run in real time on a standard CPU, without requiring depth sensors or GPU support. Fig. 3 illustrates the real time recognition stage, where the POINT hand gesture is detected from the live webcam stream and mapped to the corresponding high level swarm command (Form Line). The on screen overlay provides instant visual confirmation of the detected gesture and the command selected by the classification logic.

Fig. 3. Real time detection of the POINT gesture mapped to the Form Line swarm command.

3.3 Communication Topology and Network Architecture

The proposed system uses a centralized Star Network Topology to maintain scalable communication between the vision processing unit and the robot agents. The central node i.e. Vision Laptop connects to a standard 2.4 GHz Wi Fi router, which serves as the gateway for the swarm. The centralized star network topology and UDP broadcast based command dissemination as adopted in this work are illustrated in Fig. 4.

To minimize transmission latency, command propagation is implemented using UDP Broadcasting. Unlike connection oriented protocols (e.g., Transmission Control Protocol), which require handshake verification for every packet, UDP allows the central controller to transmit a single state packet to the broadcast address on a designated port. All robotic agents operate as listening nodes on this port, ensuring that

the entire swarm receives the control signal simultaneously. To handle packet loss and network jitter, a fault tolerant latching state strategy is employed at the robot level.

The Fault Tolerance Strategy addresses network jitter and packet loss through a "latching state" control logic. Each robot agent maintains its active kinematic state based on the last valid command received. In the event of a dropped packet, the agent continues its current trajectory until the next broadcast cycle updates the state. Given the high broadcast frequency (>30 Hz), temporary packet loss results in negligible drift, rendering the system robust against minor network delays without requiring complex synchronization overhead.

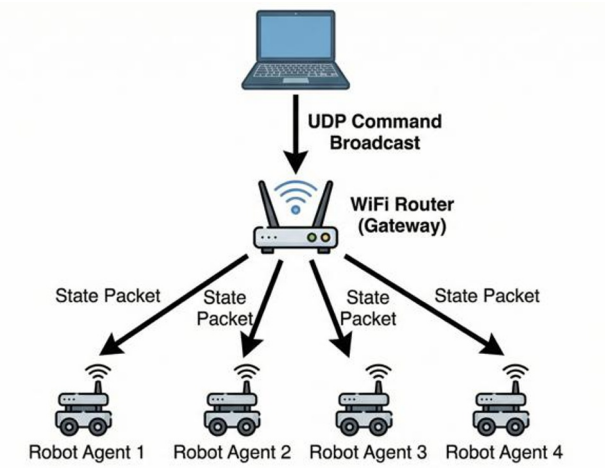

Fig. 4. Communication topology illustrating the centralized Star Network architecture.

4 Experimental Results

The vision pipeline's performance and dependability were assessed through testing with specialized software testing equipment. The system was tested on a standard laptop which had an Intel Core i5 processor and a 720p webcam, while GPU acceleration remained disabled. The system demonstration used this particular setup to show its operational capability on affordable equipment that consumers typically use.

4.1 Latency and Stability Analysis

Real time performance is critical for safety in Human Swarm Interaction [1]. High latency can lead to delayed command transmission, potentially causing robot collisions. The end to end processing time (from frame capture to command generation) is measured over a continuous stream of 500 frames.

Stability was defined as the percentage of frames where the detected gesture remained consistent while the user held a static pose. This metric is important for filtering out sensor noise.

Table 1. Real time gesture classification performance for the implemented command set

Gesture	*Stability (%)*	*Avg. Latency (ms)*	*FPS(Approx)*
POINT (Form Line)	97.93	32.4	30
VICTORY (Split)	98.66	33.1	30
PALM (Stop)	100.00	31.8	31
FIST (Wait)	95.07	32.2	31

As shown in Table 1, the system achieved an average processing latency of approximately 32 ms, translating to a smooth frame rate of 30 FPS. Notably, the "PALM" gesture, which triggers the emergency "STOP" command, exhibited 100% stability. This ensures that safety critical halt commands are robust against sensor noise.

The system response time is analyzed using an analytical model, while the vision pipeline latency was experimentally measured at 32.4 ms, the total End to End System Response Time (T_{vision}) is a critical theoretical metric for safety. The system response time is modelled analytically in Eq. (7) by summing the measured vision latency with standard hardware constraints:

$$T_{total} = T_{vision} + T_{network} + T_{actuation} \tag{7}$$

1. Vision Latency (T_{vision}): As experimentally validated in Table 1, the MediaPipe pipeline averages 32.4 ms.
2. Network Latency ($T_{network}$): Based on standard performance metrics for UDP broadcasting over 2.4 GHz Wi Fi, the transmission latency is estimated at 45 ms.
3. Actuation Latency ($T_{actuation}$): The processing overhead for a standard microcontroller (e.g., ESP32) to decode packets and trigger motor drivers is negligible, conservatively modeled at <10 ms.

$$T_{total} \approx 32.4\ \text{ms} + 45\ \text{ms} + 10\ \text{ms} \approx 87.4\ \text{ms} \tag{8}$$

As calculated in Eq. (8), this analytical model confirms that the proposed architecture falls well within the 200 ms threshold required for real time interaction [1], validating the feasibility of the vision based approach.

4.2 Accuracy Evaluation

To assess the reliability of the system, an accuracy experiment was conducted consisting of 20 independent trials for each gesture. A trial was recorded as “Correct” only if the system stabilized on the intended command within 2 seconds.

Table 2. Validation results of the gesture classifier for each gesture.

Gesture	Trials	Correct Detections	Accuracy (%)
POINT (Form Line)	20	20	100
VICTORY (Split)	20	20	100
PALM (Stop)	20	19	95
FIST (Wait)	20	18	90

The results in Table 2 demonstrate an overall average accuracy of 96.25%. The "POINT" and "VICTORY" gestures achieved perfect recognition due to the distinct geometric difference between an extended index finger and a closed fist. The "FIST" gesture had slightly lower accuracy (90%) primarily due to self-occlusion; when the hand is tightly closed, the MediaPipe landmarks for fingertips can occasionally be obscured. However, this accuracy is well within the acceptable margin for general swarm control tasks. The proposed marker-less interaction approach can also be adapted to other robotic platforms, including spatial parallel manipulators, where tolerance induced mechanical errors and parasitic motions need to be compensated [9], [10].

5 Conclusion

This work demonstrated a low-cost, vision based Human–Swarm Interaction framework that enables real time, marker-less gesture control using a standard webcam and MediaPipe hand landmarks. By transforming 21 landmarks into a normalized 42 dimensional feature vector and applying deterministic geometric rules, the system achieves reliable real time performance on consumer hardware (≈32 ms per frame, ~30 FPS) with high recognition accuracy (≈95%) and 100% stability for the safety critical STOP command. A UDP broadcast based star network topology ensures scalable and simultaneous command dissemination, while a latching state control strategy provides robustness against packet loss and network jitter. An analytical end to end response time model estimates a total system latency of approximately 87 ms, confirming that the proposed architecture satisfies real time safety requirements for Human–Swarm Interaction. Overall, the results show that practical, scalable, and delay tolerant gesture driven swarm control is achievable without specialized sensing or communication infrastructure, making the platform well suited for educational and resource-constrained research environments.

Acknowledgments. The authors gratefully acknowledge the support and guidance provided by the Department of Mechanical, COEP TU, during the preparation of this manuscript.

Disclosure of Interests. The authors declare no competing interests.

References

1. Kolling, A., Walker, P., Chakraborty, N., Sycara, K., Lewis, M.: Human interaction with robot swarms: A survey. IEEE Transactions on Human Machine Systems 46(1), 9--26 (2016)
2. Şiean, C., Grădinaru, C.: Opportunities and challenges in human swarm interaction: A systematic review. International Journal of Advanced Computer Science and Applications (IJACSA) (2023)
3. Alonso Mora, J., Breitenmoser, A., Rufli, M., Siegwart, R., Beardsley, P.: Gesture based human multi robot swarm interaction. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2015)

4. Suresh, S., Martinez, S.: Gesture based human swarm interactions for formation control using interpreters. arXiv preprint arXiv:1803.01104 (2018)
5. Krátký, V., Silano, G., Vrba, M., Papaioannidis, C., Mademlis, I., Penicka, R., Pitas, I., Saska, M.: Gesture controlled aerial robot formation for human swarm interaction in safety monitoring applications. IEEE Robotics and Automation Letters 10(8), 8244--8251 (2025)
6. Wijaya, R. S., Mubarak, M., et al.: Real time hand gesture control of a quadcopter swarm in simulation using ROS and MediaPipe Hands. Journal of Artificial Intelligence and Computing (JAIC) (2025)
7. Kakish, Z., Vedartham, A., Berman, S.: Towards decentralized human swarm interaction by means of sequential hand gesture recognition. arXiv preprint arXiv:2103.06847 (2021)
8. Wameed, A. A., Alkamachi, A. A.: Hand gesture robotic control based on computer vision. International Journal of Intelligent Systems and Applications in Engineering (IJISAE) (2024)
9. Jain, A., Jawale, H. P.: Study of the effects of link tolerances to estimate mechanical errors in 3 RRS parallel manipulator. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 236(3), 1598--1615 (2022)
10. Jain, A., Jawale, H. P.: Investigation of parasitic motion in 3 RRS parallel manipulator. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 238(14), 6964--6976 (2024)