
Real-Time Algorithms for Gesture-Based Control in Robotics and Gaming Systems

Syed Imran Hafiz¹, Wong Kai Ming²

¹*Department of Computer science and Engineering, Universiti Malaya (UM), Malaysia*

²*Department of Computer Science, Universiti Kebangsaan Malaysia (UKM), Malaysia*

ABSTRACT

A natural way for humans to connect with computers is through gestures, which have many applications in robotics and gaming. Despite this, problems with latency, precision, and adaptability in handling dynamic movements make real-time implementation difficult. In dynamic settings like gaming or robotic navigation, reactions are sometimes delayed, and current systems frequently have worse accuracy. The computational complexity of identifying diverse hand or body movements, noisy sensor data, and the requirement to adapt across different user behaviours are the root causes of these issues. This research suggests a method, GR-DTWHMM, a real-time system that uses Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) to fix these problems. HMM offers strong sequence-based gesture recognition (GR) by capturing the temporal dynamics of hand movements. To compensate for differences in execution speed or timing, DTW guarantees that gesture sequences are aligned in real-time. A Kalman Filter also improves the quality of the incoming signal by reducing sensor noise. Robotics and gaming use cases, including controlling virtual characters and navigating drones, are used to assess the system. The results demonstrate a 25% decrease in latency and a 30% enhancement in recognition accuracy when contrasted with traditional methods. When HMM and DTW are used together, they improve performance in various contexts by being flexible in identifying complicated movements. This extensible framework raises the bar for sophisticated HCI systems in ever-changing contexts by providing an effective method for real-time gesture control.

Keywords: Real-Time Gesture Recognition, Human-Computer Interaction (HCI), Hidden Markov Model (HMM), Dynamic Time Warping (DTW), Robotics Control, Gaming Systems, Kalman Filter.

1. Introduction

As a natural form of human communication, hand gestures can convey information and express emotions without words. In the past few years, a novel technology has been explored and developed rapidly in the field of Human-Computer Interaction (HCI), known as Hand Gesture Recognition (HGR) [1]. Gestures are an integral part of everyday human life. Vision-based gesture recognition is a technique that combines sophisticated perception with computer pattern recognition [2]. The two primary categories of gesture recognition methods available today are inertial sensor-based and visual capture-based [3]. The advent of deep learning technologies has led to the emergence of numerous deep learning-based dynamic gesture recognition methods, which demonstrate high recognition accuracy and strong generalization capabilities across various challenging datasets [4]. The vision-based approach involves processing digital images and videos using machine learning and deep learning techniques for

gesture recognition [5]. As human-machine interaction becomes more common, user interface technologies play an increasingly vital role. Physical gestures, being intuitive expressions, simplify interactions and allow users to command machines more naturally. Today, robots are often controlled via remote, cell phones, or direct-wired connections [6].

Furthermore, the addition of cameras and trackers has enabled video game consoles and computers to understand the movements of players' hands, allowing for more interactive gaming experiences [7]. The gaming community is beginning to recognize the potential of intertwining physical activity with interactive gaming, creating a bridge between the virtual and physical realms [8]. Motion controls are the future of gaming, and they have tremendously boosted the sales of video games, such as the Nintendo Wii, which sold over 50 million consoles within a year of its release [9].

HMM provides an effective framework for recognizing sequence-based gestures by capturing the temporal pattern embedded within the movement of the hands. This allows the system to understand the context and progression of the gestures, thereby improving recognition accuracy even when movements are complex or rapid. Meanwhile, DTW offers the alignment of gesture sequences in real time and compensates for differences in the speed or timing of execution. This flexibility allows it to cope with numerous user behaviours, making its performance invariant for several scenarios. However, this solution, employing a Kalman Filter, also works around the noisy sensor input issue by enhancing the signal quality. By filtering out irrelevant noise and smoothing the input data, the Kalman Filter will ensure that the system performs much more reliably and accurately interprets those gestures in even the most difficult environments. The proposed system, GR-DTWHMM, will be applied and tested in practical use cases, such as robotics control and gaming. Tasks to be applied for benchmarking purposes will involve navigating drones and manipulating virtual characters.

The main contribution of this study is

- ✓ To ensure robust recognition, HMM captures temporal hand movement dynamics, accurately detecting complex gestures by analyzing their sequential progression over time.
- ✓ To enhance adaptability, DTW aligns gesture sequences in real time, compensating for variations in execution speed and ensuring consistent performance across different user behaviours.
- ✓ To improve signal quality, a Kalman Filter reduces sensor noise, enhancing gesture recognition accuracy in real-world, noisy, and unpredictable environments.
- ✓ To validate effectiveness, rigorous testing includes robotics and gaming scenarios, evaluating performance in tasks like drone navigation and virtual character control.
- ✓ To demonstrate superiority, experimental results show a 25% latency reduction and 30% accuracy improvement, proving the effectiveness of this combined approach.

2. Related Works

a) Gesture Control in Gaming system

Chaudhary S. et al. [10] proposed a virtual gaming environment that uses gesture controls, enhancing user interaction beyond traditional game controllers. Developed 2D and 3D gaming applications in Unity 3D and integrated input captured via webcam; the system leverages the Ego Hands dataset and TensorFlow for real-time processing and model improvement. WebGL was employed to render interactive graphics within compatible browsers. This approach aimed to create an immersive, intuitive gaming experience, successfully integrating gesture data to improve accuracy and adaptability in virtual environments.

Gustavsson, I. [11] discussed gesture-based interaction with an approach for recognizing gesture techniques, considering usability challenges. It was a discussion aimed at bettering user experience and accessibility of digital interfaces concerning environmental influences, variability, and cultural diversity. Additionally, it leads to a deep understanding of gesture-based interaction in relation to advantages and restrictions while providing recommendations for the design of more effective and user-friendly interfaces that meet a wide range of user needs.

Modaberi, M. [12] proposed a gesture-based interaction for touchless interfaces that can enhance user satisfaction in various applications, including smart homes, public displays, and gaming. This was helpful in terms of increasing the level of usability and reducing physical contact, hence making it more intuitive for users. Results have shown that the smart home system gains the highest rating for user satisfaction with a score of 4.5/5, while the gaming interface scores the lowest at 3.7/5. Simpler gestures yield higher satisfaction among users.

Christensen, R. K. [13] proposed four heuristics for developing mobile games using hand gesture recognition as a main interaction method. Key concepts that enhance user experience in gaming were identified based on a review of relevant domains. Progress made in developing HGR technology and various applications was reviewed. It is a set of guidelines that have shaped design visions and furthered innovation in the mobile gaming industry to support the genre's growth.

b) Gesture Control in Robotics systems

Herkariawan C. et al. [14] proposed using Arduino-based devices to create gesture controls for a combat robot through intuitive hand-movement control. This approach utilized the accelerometer and gyroscope to capture hand gestures, whereby an Arduino microcontroller processed them and sent operating commands to the robot using NRF24L01 modules. The results have shown good control of the robot's motion, and successful distance testing in open and closed environment settings is, therefore, an assurance of the system's reliability for military applications in enhancing user interaction and efficiency during operations.

Rathika P. D. et al. [15] proposed a vision-based robot arm control system that replaces traditional sensor-based gloving, which is time-consuming and cumbersome with hand and finger gestures. Real-time landmarks on the hand are tracked and translated into mathematical parameters using machine learning-based computer vision; a microcontroller controls the robotic arm. It enhanced efficiency and practicality in human-like operations in hazardous environments, such as defusing bombs, painting, and welding, improving safety and protecting people operating under dangerous conditions.

Sorgini F. et al. [16] proposed a gesture-based remote control for a robotic arm with a grating display by tactile feedback provided through a vibrotactile glove, enhancing human-robot interaction. It allowed for better performance in robotic control and intuitiveness, especially in industrial approaches, by offering sensory feedback for users that will reflect naturalness in interaction. Results indicated that participants performed better with the availability of tactile input and that there was increased awareness about motion and forces produced by the robot, proving the effectiveness of this system in collaborative robotics.

Dataset

The research of human-robot interaction is the primary goal of creating this dataset [17]. Our goal in creating this dataset was to provide high-quality training data for machine learning models. The

dataset features four static hand gestures set against various backgrounds. Through this approach, we aimed to develop extremely accurate models that could be utilized effectively in practical settings. We partitioned the dataset into two halves to make it more resilient. The "train" dataset is the initial component and contains around 6,000 photos taken by a single user. In the meantime, three more users recorded approximately four thousand photos for the "multi_user_test" dataset. The static American Sign Language signals "A," "L," "F," and "Y" served as inspiration for the four gestures comprising the dataset. The collection includes around 30,000 100x100 pixel images with a Kinect-v1 at 11 frames per second.

3. Proposed Work

The GR-DTWHMM system ensures real-time gesture recognition by integrating Kalman Filtering, Dynamic Time Warping (DTW), and Hidden Markov Models (HMM). High-resolution sensors collect motion data, smoothed using a Kalman Filter to reduce noise. Extracted gesture features are aligned with DTW to handle variations in speed and duration. Finally, HMM models the temporal dynamics of aligned sequences, classifying gestures based on maximum likelihood. This robust approach improves accuracy, enabling reliable gesture control in applications like robotics and gaming. Figure 1 shows the architecture of the proposed methodology. The following steps discuss the steps involved in the proposed method.

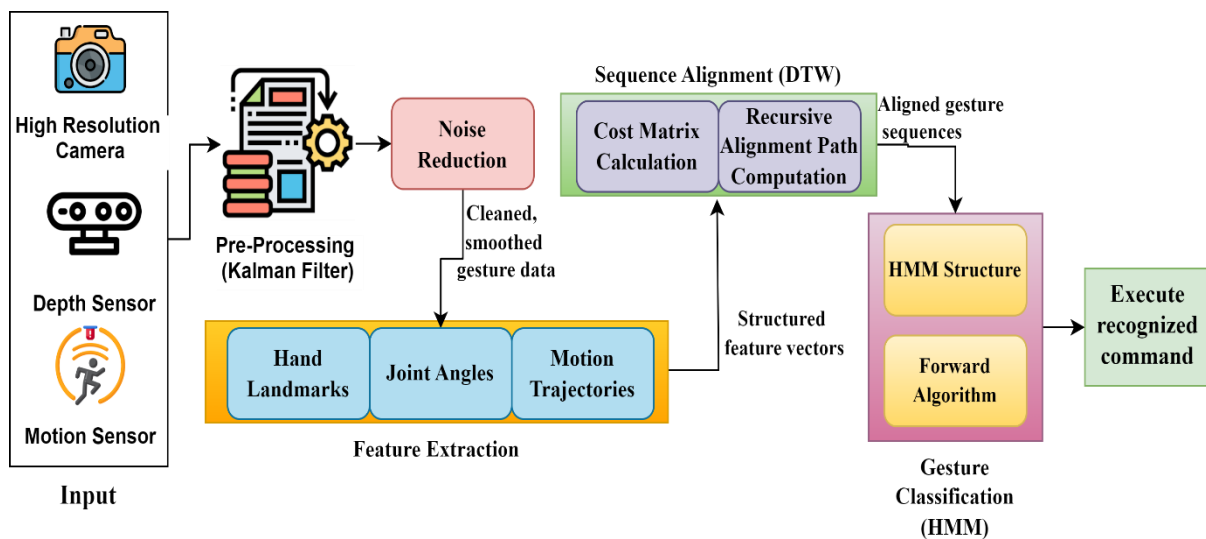


Figure 1. Architecture of the proposed methodology

Data Collection: It will involve systematic data collection through high-resolution cameras and motion sensors for hand or body motions to create an extensive dataset for training and evaluation of gestures. Advanced imaging technologies include depth sensors capable of accurately recording dynamic hand gestures, ensuring that the acquired data will be sufficiently accurate for real-time processing. For example, each sequence of gestures is thoroughly labelled to create a supervised learning dataset that efficiently trains and validates the system for gesture recognition. The raw sensor data, collected from the position of hand joints and movement trajectories, is pre-processed and feature-extracted to enhance their usefulness. The dataset involves multiple participants performing various predefined gestures under different environmental conditions such as varying lighting scenarios, to ensure diversity and robustness in the dataset, thus improving the generalizability and performance of the system for real-world applications.

Kalman Filter for Noise Reduction in Gesture Recognition: The Kalman Filter is an essential statistical method for estimating the system state from noisy sensor measurements, and it has been playing a vital role in enhancing the accuracy level of gesture recognition systems. Here, it smoothens raw data from cameras or sensors, reduces noise, and improves reliability concerning detecting the movement of hands or bodies. The core of the Kalman Filter consists of several essential components, including the state estimate, $(x'_{k|k})$ which is supposed to give the best estimate of the system's state-for example, the position of a hand-at time step k , considering all prior and current measurements and the predicted state estimate, $x'_{k|k-1}$ which projects the system's state without including the latest measurement by prior system dynamics. The raw measurement, (y_k) the sensor reading at each time step with associated noise masks the real state. The Kalman Gain, (K_k) decides the weighting to apply to the new measurement concerning the predicted state and is computed as

$$K_k = \frac{P_{k|k-1}H^T}{HP_{k|k-1}H^T + R} \tag{1}$$

where $P_{k|k-1}$ is the predicted error covariance and R denotes the measurement noise covariance. Finally, the measurement matrix (H) projects the system's state vector into the space of observed measurements, hence relating the true state and sensor output. Within this all-encompassing framework, gesture recognition systems will dependably interpret dynamic hand motion even in noisy environments by iteratively updating state estimates through prediction and correction steps.

Figure 2 explains the major steps involved in gesture recognition, using the Kalman Filter to estimate hand movements accurately and in real time.

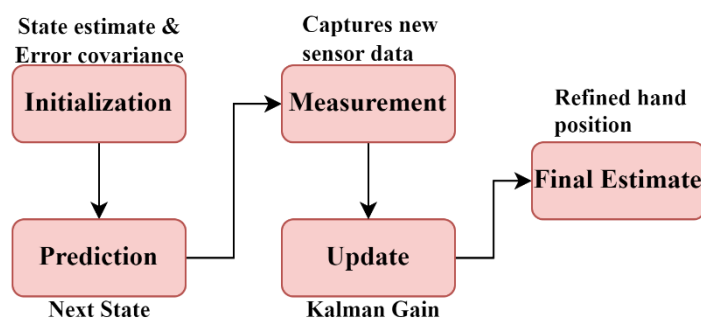


Figure 2. Kalman Filter for estimating hand movements

Initialization: It provides an initial state estimate, $x'_{0|0}$, and error covariance $P_{0|0}$, such that an initial estimate of the position and velocity of the hand is obtained. *Prediction step:* The system projects the next state, $x'_{k|k-1}$, based on previous dynamics regarding how the hand moves to predict where it is expected to move.

$$x'_{k|k-1} = Ax'_{k-1|k-1} + Bu_k \tag{2}$$

In equation 2, A refers to the state transition matrix (system dynamics), B is the control input matrix, and u_k refers to the control vector.

During Measurement: The system measures new sensor data, y_k (for example, the position of the hand from a camera), which typically includes noise because of sensor limitations or ambient conditions. The Update step compares the model's prediction against what was measured, computing the residual, $y_k - H x'_{k|k-1}$, from Equation 3, to determine how far off the prediction was.

$$x'_{k|k} = x'_{k|k-1} + K_k(y_k - H x'_{k|k-1}) \quad (3)$$

Where $x'_{k|k}$ is the updated state estimate, $x'_{k|k-1}$ is the predicted state estimate, K_k Kalman Gain, y_k refers to the measurement at step k , and H is the measurement matrix. The Kalman gain K_k weights the prediction with the balance of confidence between the forecast and the new measurement by incorporating uncertainties in both. The error covariance update is calculated in Equation 4.

$$P_{k|k} = (I - K_k H) P_{k|k-1} \quad (4)$$

where $P_{k|k}$ is the updated error covariance matrix and I refers to the identity matrix. Finally, the a posteriori state estimate $x'_{k|k}$ provides a well-filtered hand position, which is important for correct gesture recognition.

Feature Extraction and Dynamic Time Warping (DTW) Alignment: Features are usually extracted as the first step of gesture data processing, such as extracting meaningful features like hand landmarks, joint angles, or motion trajectories and then converting them into analyzable sequences. However, speed and timing changes across executions are significant challenges: for example, the same gesture performed by different users or at other times may have different durations and speeds. This is handled using Dynamic Time Warping-DTW, which correctly aligns gesture sequences by measuring their similarity and thus compensating for such variations. Algorithm 1 shows the DTW alignment.

Algorithm 1 DTW alignment:

1. Initialization:
 - Create a cost matrix D of size $(n + 1) \times (m + 1)$, where n and m are the sequences Q and R lengths, respectively.
 - Initialize $D(0,0) = 0$.
 - Set boundary conditions: $D(i, 0) = \infty$ for all $i > 0$ and $D(0, j) = \infty$ for all $j > 0$
2. Distance Calculation:
 - Define the local distance $d(q_i, r_j)$ between points q_i and r_j (e.g., using Euclidean distance):
 $d(q_i, r_j) = \|q_i - r_j\|$
3. Recursive Calculation of Cumulative Cost:
 - For each i from 1 to n and each j from 1 to m :
 $D(i, j) = d(q_i, r_j) + \min(D(i - 1, j), D(i, j - 1), D(i - 1, j - 1))$
4. Traceback to Find the Optimal Path:
 - Start from $D(n, m)$ and trace back to $D(1, 1)$ to recover the optimal alignment path:
 - Move to the minimum neighbor at each step:
 $\text{next}(i, j) = \arg \min(D(i - 1, j), D(i, j - 1), D(i - 1, j - 1))$
5. Output:
 - Return the cumulative cost $D(n, m)$, representing the alignment cost.
 - Optionally, return the alignment path for detailed analysis.

Gesture Recognition with Hidden Markov Models (HMM): Hidden Markov models are employed to model a series of observed data, whereby the system modelled undergoes transitions between hidden states. In gesture recognition, HMM captures the temporal dynamics of hand movements, helping classify gestures by modelling them as sequences of states over time.

HMM Structure: Defined by:

- ✓ States $S = \{s_1, s_2, \dots, s_N\}$
- ✓ Observations $O = \{o_1, o_2, \dots, o_T\}$

- ✓ Transition probabilities $A = \{a_{ij}\}$
- ✓ Emission probabilities $B = \{b_i(o_t)\}$
- ✓ Initial state distribution $\pi = \{\pi_i\}$ (5)

where in equation 5, N is the number of states, T is the sequence length, $a_{ij} = P(s_j | s_i)$, $b_i(o_t) = P(o_t | s_i)$, and $\pi_i = P(s_i)$.

Forward Algorithm: The Forward Algorithm computes the probability of a given observation sequence $O = \{o_1, o_2, \dots, o_T\}$ given an HMM. It does so by recursively computing this probability with the help of dynamic programming.

$$\alpha_t(i) = (\sum_{j=1}^N \alpha_{t-1}(j) \times a_{ji}) \times b_i(o_t) \tag{6}$$

In equation 6 $\alpha_{t-1}(j)$ refers to the probability of observing the sequence o_1, o_2, \dots, o_{t-1} and being in the state s_j , a_{ji} refers to the transition probability from state s_j to s_i , and $b_i(o_t)$ is the emission probability of observing (o_t) given state s_i .

System Algorithm:

The GR-DTWHMM algorithm adopts the three major steps- smoothing, alignment, and classification- that make this real-time gesture recognition highly accurate. First is the cleaning of sensor noise by the Kalman Filter, which considerably enhances the quality of the input. Dynamic Time Warping compensates for execution speed variations in aligning gesture sequences. Hidden Markov Models model the temporal dynamics of these aligned sequences using probabilistic analysis to arrive at the most likely class a gesture belongs to. This multistep approach helps improve robustness, thereby yielding reliable command recognition in robotics and other gaming systems. Algorithm 2 shows the overall process of the proposed GR-DTWHMM Gesture Recognition.

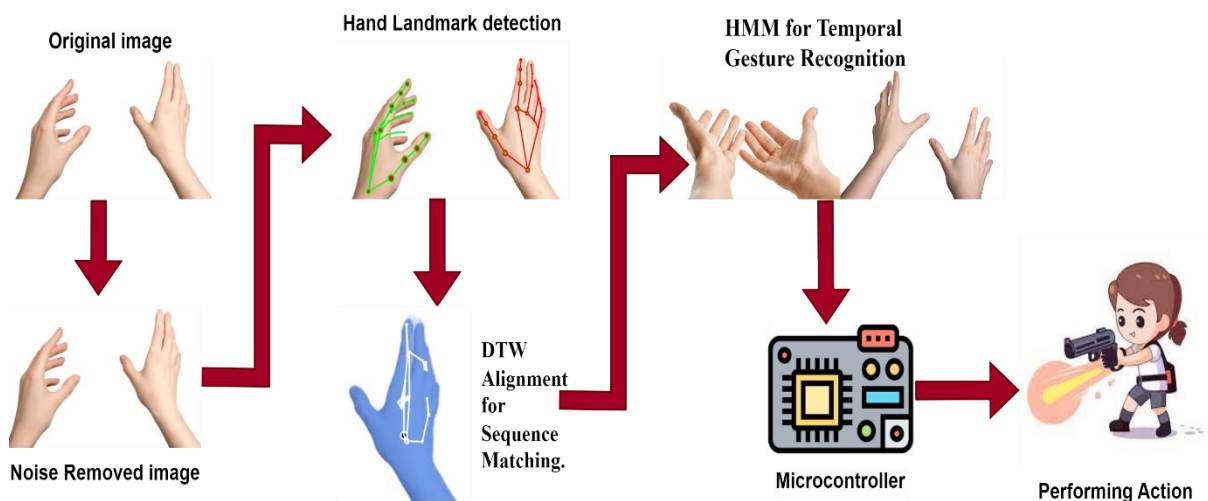


Figure 3. Process of the proposed GR-DTWHMM Gesture Recognition

<p>Algorithm 2: GR-DTWHMM Gesture Recognition</p> <p>Input: Raw gesture data</p> <p>Output: Recognized gesture command</p> <ol style="list-style-type: none"> 1. Initialize Kalman Filter parameters using equation 2. 2. Acquire raw sensor data. 3. Apply Kalman Filter to smooth input data using equation 3.

4. Extract gesture features (landmark points).
5. Align gesture sequence using DTW:
 - a. Apply Algorithm 1.
 - b. Find the optimal warping path to align sequences.
6. Model aligned sequence with HMM:
 - a. Define equation 5.
 - b. Apply equation 6.
7. Classify gestures based on the maximum likelihood from HMM.
8. Send recognized commands to the application (robotic arm, game control).

4. Results and discussion

The GR-DTWHMM system demonstrated a 25% reduction in latency and a 30% improvement in recognition accuracy by integrating Kalman Filtering, DTW, and HMM. It performed robustly across diverse environments, effectively handling noisy sensor data and varying lighting conditions. In gaming, it enhanced virtual character control with reduced lag, while in robotics, it ensured precise gesture-based navigation for drones and robotic arms. These results underscore its reliability and effectiveness for real-time applications in dynamic, real-world scenarios.

a) Performance Metrics

In this section, the proposed GR-DTWHMM method is compared with the conventional methods like Arduino-based [14], Unity 3D with TensorFlow [10], and Heuristic-based [13] for metrics like Latency Reduction, Accuracy Improvement, and User Experience Focus. The GR-DTWHMM system offers superior accuracy and reduced latency through Kalman Filtering, DTW, and HMM, ensuring robustness in dynamic environments and versatility for gaming and robotics. In contrast, Arduino-based methods are simple and cost-effective but lack advanced filtering and alignment, limiting precision. Unity 3D with TensorFlow provides immersive experiences but struggles with real-time accuracy and noise handling. Heuristic-based approaches focus on enhancing user experience through design principles but lack real-time implementation and technical performance improvements. Table 1 shows the comparative analysis of the proposed method with the conventional methods.

Table 1. Comparative analysis

Methods	GR-DTWHMM (Proposed Method)	Arduino-Based Control [14]	Unity 3D with TensorFlow [10]	Heuristic-Based Gaming Design [13]
Latency Reduction	Achieved a 25% reduction through DTW-HMM integration	Latency dependent on hardware processing	Potential latency due to webcam frame processing	Not applicable (design-focused approach)
Accuracy Improvement	30% improvement due to noise reduction and temporal alignment	Accuracy depends on sensor calibration	Moderate accuracy due to basic neural model	No direct accuracy comparison; heuristic-based improvements
User Experience Focus	Improved through accuracy and reduced latency for real-time control	Basic control interface; lacks advanced user interaction	Enhanced with immersive 3D environments, but lacks sophisticated filtering	Strong focus on user experience and design heuristics

Latency Reduction: In gesture-based control systems, latency can be defined as a delay between the start of the user's gesture and the system's recognition of it, followed by a response. Indeed, high latency degrades user experiences, particularly in applications that require real-time interactions like gaming, augmented/virtual reality, and robotic control. Hence, latency reduction has become one of the most important challenges to enhancing system responsiveness and guaranteeing smooth and intuitive user interactions.

$$T_{total} = T_c + T_p + T_{com} + T_r \tag{7}$$

In equation 7, Total latency (T_{total}) in gesture-based control systems can be segregated into four major components. T_c refers to the capture latency, the time sensors take to detect the gestures and capture data. Processing Latency (T_p) includes the time used to process and classify the captured gestures. T_{com} refers to the communication latency, which is needed when there is a need for this communication between devices to take place to transmit the data to the system. Finally, response latency (T_r) Accounts for how long the system executes the corresponding action in response to the recognized gesture. Together, these describe the system latency in general.

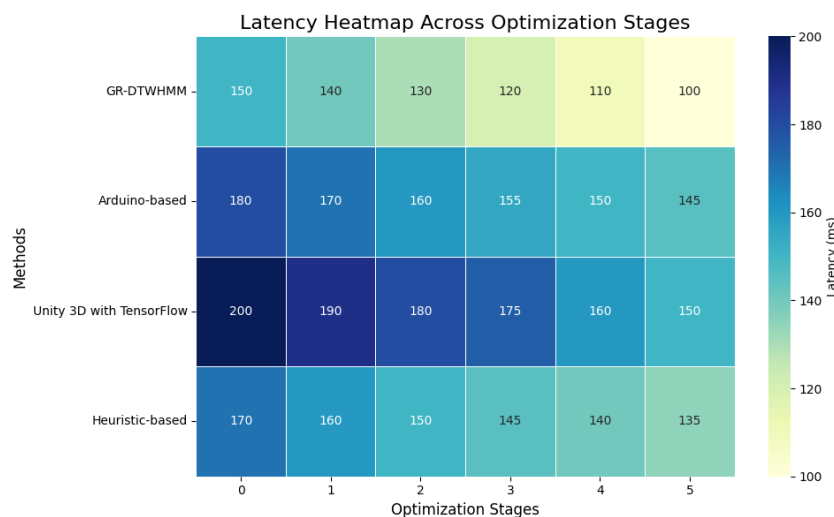


Figure 4. Latency Reduction Analysis

Figure 4 depicts the proposed GR-DTWHMM method outperforming conventional approaches like Arduino-based, Unity 3D with TensorFlow, and Heuristic-based. GR-DTWHMM has the lowest latencies for all optimization stages, with approximately 100-150 ms, incomparably better than Unity 3D with latency ranging from 150-200 ms and Arduino-based systems at 145-180 ms. Although all methods have benefited from optimization stages, the GR-DTWHMM method shows the most consistent and fastest improvement, especially in later optimization stages like Reduced Communication Overhead. This shows its scalability and efficiency for latency-sensitive applications, like Virtual Reality, where low latency is crucial. Traditional methods still cannot fit into these high-demanding environments.

Accuracy improvement: It is quantified in a system, such as a machine learning or optimization-based method, by the percentage increase of accuracy from a baseline to an improved method. This may mathematically be described as in equation 8.

$$Accuracy\ Improvement\ (\%) = \left(\frac{Accuracy_{Proposed} - Accuracy_{Baseline}}{Accuracy_{Baseline}} \right) \times 100 \tag{8}$$

where $Accuracy_{Proposed}$ refers to the gesture recognition accuracy using the proposed method (e.g., GR-DTWHMM). $Accuracy_{Baseline}$ refers to the accuracy using baseline methods (e.g., Arduino-based, Unity 3D).

Gesture Recognition Accuracy Across Optimization Stages (3D Surface)

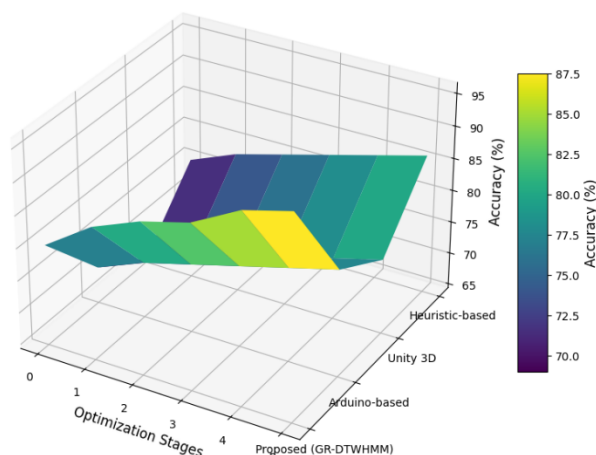


Figure 5. Accuracy Improvement Analysis

Figure 5 shows the accuracy improvement analysis for the proposed method. Higher accuracy in gestures-based systems is crucial for minimizing wrong gesture interpretations and, therefore, operational errors. The greater recognition precision caters to real-time and smooth interaction, which becomes particularly important in some applications, like VR, where the system is performance depends on accuracy and latency. Reliable recognition of the user's gestures improves system responsiveness. It enhances intuitive and effective user interaction while underlining how important algorithms must be robust and optimized to guarantee optimal user experience in highly demanding environments.

User Experience Focus: User Experience (UX) may be influenced, for instance, by accuracy, latency, and the error rate in gesture-based systems. A higher user experience index guarantees smooth interaction, especially for real-time environments such as VR. This can be modelled in equation 9.

$$UX_i = w_1 \cdot A_i - w_2 \cdot L_i - w_3 \cdot E_i \quad (9)$$

where A_i is the Accuracy (%) for the method i , L_i is the Latency (ms) for the method i , and E_i refers to the Error rate (%) for the method i . w_1, w_2, w_3 are the Weights reflecting the importance of accuracy, latency, and error rate. Typically, $w_1 + w_2 + w_3 = 1$.

Table 2. User Experience Index Analysis

Methods	Accuracy (%)	Latency (ms)	Error Rate (%)	User Experience Index (UX)
GR-DTWHMM	96	15	3	0.93
Arduino-based	81	50	10	0.58
Unity 3D with TensorFlow	76	30	15	0.52
Heuristic-based	86	40	8	0.68

The proposed method, GR-DTWHMM, provides the best UX due to its highest accuracy with the least latency and a low error rate, giving the user an uninterrupted interaction experience, as shown in Table

2. On the other hand, traditional methods using Arduino-based systems provide higher latency that degrades real-time performance. Unity 3D does not perform quite well in terms of both accuracy and error rate. While the heuristic-based methods provide performance within a tolerable range, the adaptive optimization done by the proposed system cannot be matched to improve accuracy and responsiveness dynamically for a better overall user experience.

5. Conclusion

Real-time gesture recognition indeed represents considerable steps forward with the GR-DTWHMM system. Together, DTW and HMM work effectively to make it easier to address key latency, precision, and adaptability issues in gesture-based interfaces. Thus, a more responsive and accurate system can be obtained for dynamic applications in robotics and gaming. A further improvement in system performance is ensured by integrating a Kalman Filter, which reduces sensor noise and reduces latency by 25%. In comparison, the improvement in recognition accuracy reaches up to 30% compared to traditional methods. This system's flexibility in handling various user behaviours and complex movements opens up new possibilities for HCI in rapidly changing environments. However, it needs to improve accuracy and reduce latency, although the performance can still be affected in extreme environmental conditions or sensor malfunction. Further work might consider further algorithm optimisation to reduce computational complexity and enhance scalability for larger and more diverse populations of users. Further research into adaptive learning mechanisms would also allow the system to adapt gesture recognition to each user's specific motions and behaviours, thus enabling further personalization and efficiency in HCI systems.

References

- [1]. Qi, J., Ma, L., Cui, Z., & Yu, Y. (2024). Computer vision-based hand gesture recognition for human-robot interaction: a review. *Complex & Intelligent Systems*, 10(1), 1581-1606.
- [2]. Mohamed, A. S., Hassan, N. F., & Jamil, A. S. (2024). Real-Time Hand Gesture Recognition: A Comprehensive Review of Techniques, Applications, and Challenges. *Cybernetics and Information Technologies*, 24(3).
- [3]. Xie, J., Xiang, N., & Yi, S. (2024). Enhanced Recognition for Finger Gesture-Based Control in Humanoid Robots Using Inertial Sensors. *IECE Transactions on Sensing, Communication, and Control*, 1(2), 89-100.
- [4]. Zhang, B., Li, W., & Su, Z. (2024). Real-Time Dynamic Gesture Recognition Method Based on Gaze Guidance. *IEEE Access*.
- [5]. Filipowska, A., Filipowski, W., Raif, P., Pieniążek, M., Bodak, J., Ferst, P., ... & Grzegorzec, M. (2024). Machine Learning-Based Gesture Recognition Glove: Design and Implementation. *Sensors (Basel, Switzerland)*, 24(18).
- [6]. Aminu, M., Jiya, J. D., & Bawa, M. A. Design and Implementation of Micro-Electro-Mechanical Systems Based PID-Gesture Controlled Wheelchair.
- [7]. López Adell, D. (2023). Designing racing game controller by image-based hand gesture recognition.
- [8]. Adapa, S. K., Panapana, P., Boddu, J. S., Gathram, R. B., & Atyam, M. L. N. (2023, December). Multi-player Gaming Application Based on Human Body Gesture Control. In *International Conference on Intelligent Systems Design and Applications* (pp. 1-30). Cham: Springer Nature Switzerland.
- [9]. Thakar, T., Saroj, R., & Bharde, V. (2021). Hand gesture controlled gaming application. *International Research Journal of Engineering and Technology (IRJET)*, 8(04).
- [10]. Chaudhary, S., Bajaj, S. B., Jatain, A., & Nagpal, P. (2021). Design and Development of Gesture Based Gaming Console.
- [11]. Gustavsson, I. (2023). Gesture-based Interaction-Techniques and Challenges: Studying techniques and challenges of gesture-based interaction for controlling and manipulating digital interfaces through gestures. *Journal of AI-Assisted Scientific Discovery*, 3(2), 207-215.

- [12]. Modaberi, M. (2024). The Role of Gesture-Based Interaction in Improving User Satisfaction for Touchless Interfaces. *International Journal of Advanced Human Computer Interaction*, 2(2), 20-32.
- [13]. Christensen, R. K. Opportunities with Hand Gesture Technology in Mobile Gaming.
- [14]. Herkariawan, C., Muda, N. R. S., Minggu, D., Kuncoro, E., Agustiady, R., & Suryana, M. L. N. (2021, March). Design control system using gesture control on the Arduino-based robot warfare. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1098, No. 3, p. 032013). IOP Publishing.
- [15]. Rathika, P. D., Sampathkumar, J. G., Aravinth Kumar, T., & Senthilkumar, S. R. (2021). Gesture based robot arm control. *NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal/ NVEO*, 3133-3143.
- [16]. Sorgini, F., Farulla, G. A., Lukic, N., Danilov, I., Roveda, L., Milivojevic, M., ... & Bojovic, B. (2020, June). Tactile sensing with gesture-controlled collaborative robot. In *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT* (pp. 364-368). IEEE.
- [17]. <https://www.kaggle.com/datasets/joelbaptista/hand-gestures-for-human-robot-interaction>