

GESTURE RECOGNITION SYSTEM FOR CONTROLLING IOT SYSTEMS

Maksym Ferents, Ihor Rabiichuk, Andrii Fechan

Lviv Polytechnic National University, Lviv, Ukraine

maksym.ferents.pz.2021@lpnu.ua, ihor.o.rabiichuk@lpnu.ua, andrii.v.fechan@lpnu.ua

<https://doi.org/10.23939/jcpee2025.01.018>

Abstract: The development of the Internet of Things (IoT) opens up new opportunities for creating intelligent services that enhance user interaction with surrounding devices. Modern IoT systems primarily use touchscreens and mobile applications for control; however, gesture-based methods can significantly expand their functionality. This work proposes a gesture recognition system applied to the control of IoT devices. The core of the system is the classification of finger movement trajectories using a Hidden Markov Model (HMM). The system consists of three main stages: initial hand segmentation using colour and depth information, fingertip detection based on hand contours, and the use of clustering in polar coordinates to extract dynamic features. The Baum-Welch and Viterbi algorithms are applied for training and gesture recognition, respectively. Experimental results show that the developed system is capable of classifying gestures with consideration of spatiotemporal variability with high accuracy. In particular, the average recognition rate reached 98.61 % for the training set and 93.06 % for the test data. The proposed approach demonstrates effectiveness under challenging conditions, including changes in lighting and partial occlusion of objects in the scene.

Keywords: internet of Things (IoT), gesture recognition, gesture control, colour information, depth map, Hidden Markov Model.

1. Introduction

With the development of the Internet of Things (IoT), smart devices are increasingly being integrated into everyday human life. Interaction between users and IoT devices is traditionally carried out through touchscreens or mobile applications, requiring direct physical contact or voice commands. However, the use of gesture-based interaction opens up new opportunities for contactless device control and enhances user comfort [1].

Gesture interaction is a promising technology that enables intuitive device control by tracking hand movements in space. This is particularly important for applications in smart homes, industrial automation systems, and medical devices, where reducing physical contact is critical [2]. The main challenge in this field is ensuring accurate and fast gesture recognition in real-time, which requires the application of modern computer vision techniques, machine learning, and spatial data processing [3].

This work considers a gesture recognition system designed for controlling IoT services. The proposed approach is based on the use of colour information and scene depth for hand segmentation, as well as the application of a Hidden Markov Model (HMM) for classifying fingertip movement trajectories. This allows achieving high recognition accuracy even under challenging conditions, such as varying lighting or partial occlusion of hands by other scene objects [4].

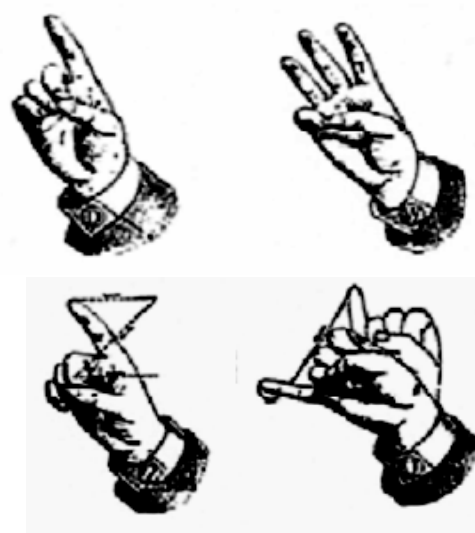


Fig. 1. The first two examples represent posture, while the other examples relate to gestures

The following sections of this article include an analysis of previous research in the field of gesture control (Section 2), a detailed description of the proposed system architecture (Section 3), methods for hand segmentation and keypoint detection (Section 4), gesture processing algorithms (Section 5), and the results of experimental testing (Section 6). Conclusions and potential directions for future research are presented in the final part of the paper.

2. Previous researches

The importance of Internet of Things (IoT) technologies has been recognized by numerous studies [1–3, 5]. Want et al. [6] emphasize that IoT enables users to monitor and control devices through internet-based

technologies. Some works focus on the use of gesture control for IoT. For example, Han and Rashid [7] proposed a system that combines voice and gesture control of IoT devices, which includes two stations: a control station and a device station. For gesture recognition, they applied contour, convex hull, and trajectory simplification algorithms.

One of the key applications of gesture recognition is sign language recognition, which allows users to interact with computers. Various methods for classifying and recognizing gestures, such as alphabets and numbers, have been presented in the literature. For instance, an Adaptive Neuro-Fuzzy Inference System (ANFIS) was used for recognizing Arabic sign language [8]. To simplify segmentation, coloured gloves were employed, helping to extract the most relevant gesture features. Another approach, proposed by Handouyahia et al. [9], utilized neural networks for the classification and recognition of the International Sign Language (ISL). The main advantage of this approach is the ability to accurately train and test the extracted gesture features.

In 3D hand posture recognition, an important factor is the use of Elliptical Fourier Descriptors (EFD), which enable effective gesture recognition [10]. Liksar and Shirany [11] applied Fourier coefficients to analyze hand shapes for gesture recognition, while Freeman and Roth [12] used hand orientation histograms for alphabet symbol classification, which helped reduce the probability of misclassification.

Unlike previous studies, this work proposes a gesture recognition system for controlling IoT services. It is based on the recognition of alphabet symbols and numbers using a Hidden Markov Model. Moreover, in contrast to the approach by Han and Rashid [7], the proposed system is capable of recognizing gestures even when the background contains colours similar to skin or facial tones. This is achieved using a Gaussian Mixture Model (GMM) trained on both skin and non-skin data samples. Experimental results show that the system achieves high accuracy rates: 98.61 % for the training dataset and 93.06 % for the test dataset. Additionally, the proposed approach demonstrates superior performance when processing sequences of images under challenging conditions.

Researchers are actively exploring human-computer interaction (HCI) in the context of IoT. Gesture recognition is widely used for interpreting sign language; however, its application in controlling IoT devices remains limited. Previous studies have proposed various methods for gesture analysis, including neural network approaches, orientation histograms, and adaptive clustering algorithms.

3. Architecture of the system

The proposed gesture recognition system is based on a multi-layered architecture that ensures effective interaction between the user and IoT devices. This architecture

integrates various technologies that enable the analysis of hand movements in space and their interpretation for device control.

At the device level, various IoT devices equipped with built-in sensors, cameras, and wireless communication modules are deployed. These devices perform the initial data collection about the surrounding environment and transmit the information for further processing. Cameras capture hand movements, while sensors can record additional parameters such as acceleration or pressure.

The mobile network layer is responsible for maintaining continuous communication between IoT devices and cloud or local data processing services. Technologies such as Wi-Fi, Bluetooth, Zigbee, or mobile networks (3G/4G/5G) may be used for this purpose. A critical feature is low data transmission latency, which enables real-time signal processing.

At the gesture recognition layer, the collected data is analyzed, video streams are processed, and user gestures are classified. The first step is fingertip detection, which is performed based on colour and depth information analysis. Next, feature extraction methods are applied to describe the hand movement trajectory in polar coordinates. The resulting data is then fed into a Hidden Markov Model (HMM) algorithm, allowing gestures to be recognized with high accuracy. The final component is a service invocation module, which interprets the recognized gesture and sends the corresponding command to the IoT device.

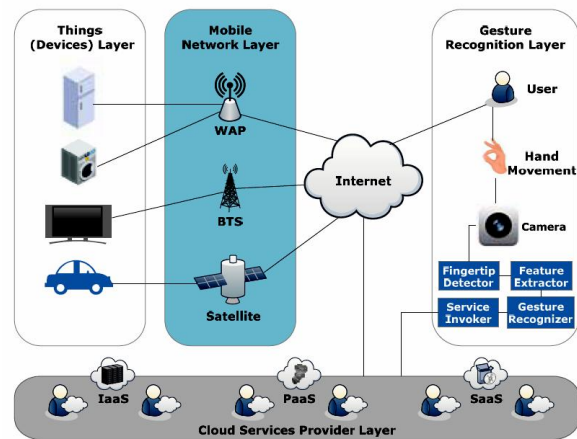


Fig. 2. Architecture of a gesture recognition system for gesture control in IoT services

The cloud layer provides the necessary computational resources for analyzing large volumes of data and long-term storage of information. Depending on the requirements, various cloud service models, such as Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS), may be employed. For example, a database of recognized gestures can be stored in the cloud for further training and system improvement.

4. Segmentation and detection of the tip of the finger

To detect fingertips, we use a standard laptop camera and skin colour analysis. The main task is to separate the hand from the background correctly and find the contours of the fingers.

Segmentation of the arm

To determine the area of the hand, we analyse colour information. We use the YCbCr colour space, where the components (Cb, Cr) help to find the skin, and brightness (Y) is ignored to reduce the influence of lighting. We apply a Gaussian mixture model (GMM) that learns from skin and background samples, allowing us to find the hand area more accurately even in complex backgrounds.

Determination of fingertips

After finding the contour of the hand, we look for curved points – these are the fingertips. We use the k-curvature method, which analyses changes in the direction of the contour [15]:

$$k = \sum_{i=-n/2}^{n/2} \frac{|p_{i+n/2} - p_{i-n/2}|}{d},$$

where p_i are the points of the contour, and d is the distance between the first and last points of the segment.

If the curvature value exceeds the threshold (typically between 1 and 4), the point is marked as a fingertip. Fig. 3 shows fingertip detection using the method of clustering curvature peaks and valleys. To reduce detection errors, an additional normalization step is applied, where the distance between the palm center and the fingertip is calculated. This makes it possible to more effectively distinguish actual fingertips from false detections.

5. Identifying features

Gesture recognition largely depends on the correct extraction of meaningful features from raw hand motion data. The extracted features should effectively represent

the gesture while maintaining computational efficiency. In this study, we employ a feature extraction method based on polar coordinates, which provides a robust representation of the trajectory.

To extract the dynamic features of a gesture, we first represent the trajectory of the fingertip motion in polar coordinates. Let the set of fingertip positions in Cartesian coordinates be given as (x_t, y_t) . These can then be transformed into polar coordinates (ρ_t, ϕ_t) , where ρ_t is the radius vector and ϕ_t is an angle:

$$\rho_t = \sqrt{x_t^2 + y_t^2}, \quad \phi_t = \tan^{-1} \left(\frac{y_t}{x_t} \right).$$

The polar coordinates obtained are used to construct a set of features for each gesture:

$$F_c = \{(\rho_{c1}, \phi_{c1}), (\rho_{c2}, \phi_{c2}), \dots, (\rho_{cT-1}, \phi_{cT-1})\},$$

where F_c represents a sequence of polar coordinate pairs (ρ, ϕ) for a given gesture. Similarly, for the gesture recognition process, we define:

$$F_{sc} = \{(\rho_{sc1}, \phi_{sc1}), (\rho_{sc2}, \phi_{sc2}), \dots, (\rho_{scT-1}, \phi_{scT-1})\},$$

where F_{sc} – this is a set of features obtained for the gesture classification model.

The use of polar coordinates has several advantages. First, they provide invariance to translation and scaling, since radial and angular measurements remain unchanged regardless of changes in position in the original coordinate space. Second, this method allows for effective clustering of similar gestures even in noisy conditions using the k-means clustering algorithm. This is confirmed in publications [22, 23], which demonstrate that polar coordinates reduce data variability during rough manipulations.

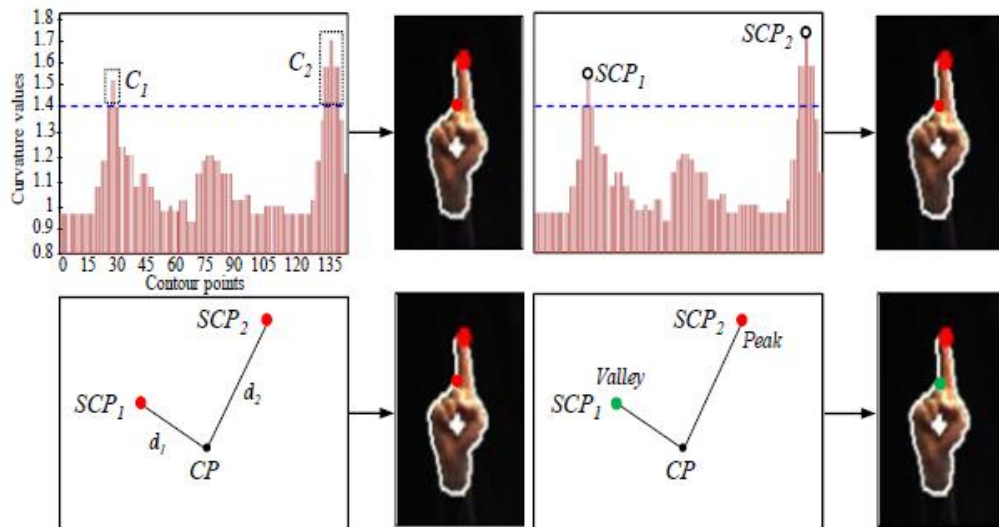


Fig. 3. Detection of peaks and valleys

Gesture Representation and Segmentation

To improve accuracy, we perform preprocessing of the data, including hand segmentation and fingertip position detection. The detected fingertip is tracked over time, forming a continuous trajectory that serves as the basis for gesture recognition. To smooth the trajectory and eliminate small fluctuations caused by sensor noise, a moving average filter is applied, which reduces random variations [23].

The resulting trajectory is projected into the feature space defined by the sets F_c and F_{sc} . Since gesture recognition requires both spatial and temporal analysis, a Hidden Markov Model (HMM) is employed for classification. The HMM parameters are optimized using the Baum–Welch algorithm, while the Viterbi algorithm is applied to determine the most likely sequence of gestures based on the observed feature vectors [24].

Illustration of the Feature Extraction Process

Fig. 4 demonstrates the transition from Cartesian to polar coordinates. The left part of the figure (a) shows the

original gesture trajectory in the x, y -space. The central part (b) presents the corresponding trajectory in the polar coordinate space (ρ, φ) which reduces dependency on the absolute position. The right part (c) illustrates the transformed $\rho\varphi\omega$ feature space, which incorporates additional temporal information to enhance classification accuracy.

6. Classification based on the hidden markov model

During the classification process, the obtained symbols are correlated with the corresponding classes. The main stages of this process include model training and testing of the obtained results. The Baum–Welch algorithm [24] is used to train the parameters of the hidden Markov model (HMM). During testing, the obtained gesture is compared with the gesture database stored in the cloud environment to determine its corresponding class. Recognition is based on the Viterbi algorithm [24], which selects the model with the highest probability of gesture matching (Fig. 5)

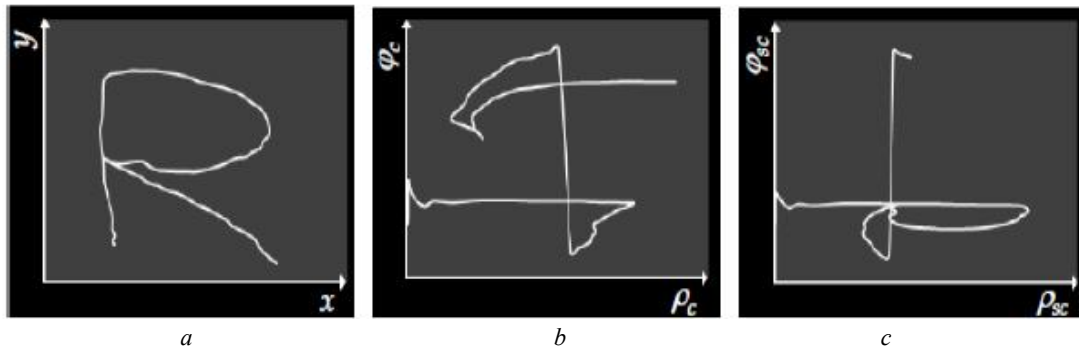


Fig. 4. The process of transition from Cartesian coordinates to polar coordinates

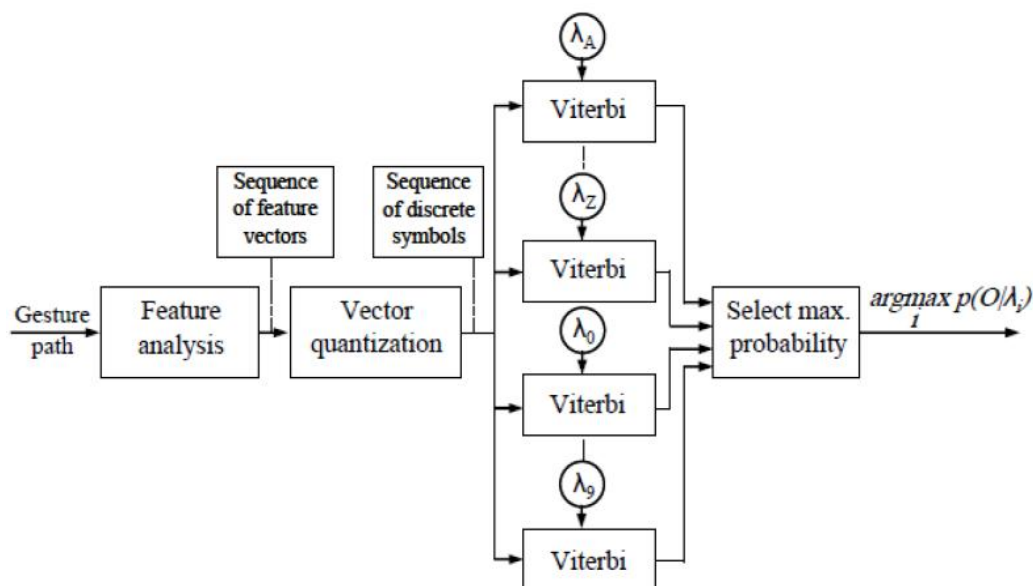


Fig. 5. Block diagram of the process of recognising isolated gestures using the Viterbi algorithm

Before starting training, it is necessary to determine the number of HMM states for each gesture. The number of states depends on the complexity of the gesture, since each gesture is broken down into separate segments corresponding to a specific state. For example, two states are sufficient for the ‘L’ gesture, while six states are required for ‘E’ and four for ‘3’. If the number of states is too large, the model may become too specific (overfitting), which will impair the generalisation of new data. To prevent this problem, regularisation, cross-validation, and early termination of training techniques are used [25].

Before running the Baum-Welsh algorithm, it is necessary to set the initial HMM parameters: the transition probability between states (matrix A), the emission probability of observations (matrix B), and the initial state probabilities (vector π). For this purpose, the Left-Right Banded Model (LRB) [24] is used, which allows changing the state only in the forward direction or remaining in the same state. The structure of the transition matrix A looks like this:

$$A = \begin{pmatrix} a_{11} & 1-a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & 1-a_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}.$$

The diagonal elements of this matrix determine the average duration of stay in each state:

$$a_{ii} = 1 - \frac{1}{d} \quad d = \frac{T}{N},$$

where N is the number of states and T is the length of the gesture trajectory.

The matrix B describes the observation probabilities of symbols in each state. The general form of matrix B is:

$$B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \dots & b_{NM} \end{pmatrix} = \begin{pmatrix} \frac{1}{M} & \frac{1}{M} & \dots & \frac{1}{M} \\ \frac{1}{M} & \frac{1}{M} & \dots & \frac{1}{M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} & \frac{1}{M} & \dots & \frac{1}{M} \end{pmatrix},$$

$$b_{iM} = 1/M,$$

where M – the number of possible symbols in the model. The initial state is determined by the vector π :

$$\pi_1 = 1, \pi_2 = \pi_3 = \dots = \pi_N = 0.$$

The Baum-Welsh algorithm is an effective training method. Typically, a high-quality model is obtained after six to ten iterations. training continues until the changes in matrices A and B are smaller than the set threshold of

$$\varepsilon = 0,001. \quad |P^{(k+1)} - P^k| < \varepsilon.$$

This reduces the number of iterations and prevents overfitting. After the training is completed, the resulting model is capable of effectively classifying new gestures, even if they differ from those used in the training set.

It should also be noted that the performance of HMMs strongly depends on the choice of the number of states and the quality of the initial training. Using an appropriately selected number of states and well-tuned model parameters contributes to improving gesture recognition accuracy, even under conditions of partial occlusion or changes in the background environment.

7. Results of the experiments

To evaluate the effectiveness of the developed gesture recognition system, a series of experiments was conducted, including hand segmentation and tracking in complex scenes. The use of colour information together with 3D depth reduced the influence of illumination changes and occlusion of the region of interest. For this purpose, the YCbCr colour space was employed in combination with a Gaussian Mixture Model (GMM), which enabled the automatic detection of hand and face skin regions [27].

The segmentation procedure relied on the processing of colour channels (Cb and Cr), which reduced sensitivity to brightness variations. Next, the k-means algorithm was applied for clustering, followed by hand tracking using the mean-shift method, which allowed the fingertip trajectory to be determined in each frame.

To improve the accuracy of estimation, the hand colour histogram was smoothed using the Epanechnikov kernel [28]. The Bhattacharyya coefficient [18] was then used to compare the target and current hand positions, which helped reduce errors. In addition, the mean depth value of the ROI was employed to resolve the problem of occlusion between the hands and the face.

After obtaining gesture trajectories, feature clustering was performed using k-means, which allowed the motion characteristics to be correctly distributed among the corresponding classes. For training and testing, 720 and 360 image sequences were used, respectively. The average recognition accuracy reached 98.61 % for the training set and 93.06 % for the test set, demonstrating the high effectiveness of the system (Table).

Assessment of system effectiveness

Feature	Training Set (%)	Test Set (%)	General result (%)
Fc	94.42	84.73	89.58
Fsc	95.83	86.11	90.75
Fc+Fsc	98.61	93.06	95.84

As can be seen from the results, the use of combined features (Fc + Fsc) provided the best recognition results. In addition, the system demonstrated resistance to changes in lighting and partial overlap.

Further research will focus on expanding the system's capabilities to support word and sentence recognition, which will allow it to be integrated into real-world IoT control systems.

8. Conclusions

This paper looked at a gesture recognition system for controlling IoT devices based on hidden Markov models. Using segmentation methods based on colour and depth info really helped improve how accurately it could spot hands and their movements. The use of clustering algorithms and spatio-temporal characteristics ensured effective model training and high gesture classification accuracy.

The results of the experiments showed that the proposed approach demonstrates resistance to changes in lighting, partial overlaps, and other factors that can affect recognition accuracy. The average gesture classification accuracy reached 98.61% on training data and 93.06 % on test data.

Future research will focus on expanding the system's functionality to recognise more complex gestures, as well as integrating it into real IoT environments to improve the user experience of contactless device control.

References

- [1] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A Survey on Internet of Things From Industrial Market Perspective", *IEEE Access*, vol. 2, pp. 1660–1679, 2014.
- [2] A. Whitmore, A. Agarwal and L. Da Xu, "The Internet of Things – A Survey of Topics and Trends", *Information Systems Frontiers*, vol. 17, No. 2, pp. 261–274, 2015.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communications Surveys & Tutorials*, vol. 17, No. 4, pp. 2347–2376, 2015.
- [4] R. Bowden, A. Zisserman, T. Kadir, and M. Brady, "Vision Based Interpretation of Natural Sign Languages", *In Proceedings of the International Conference on Computer Vision Systems*, 2003.
- [5] R. Want, B. N. Schilit, and S. Jenson, "Enabling the Internet of Things", *IEEE Computer*, vol. 48, No. 1, pp. 28–35, 2015.
- [6] X. Han and M. A. Rashid, "Gesture and Voice Control of Internet of Things", *In Proceedings of IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1791–1795, 2016.
- [7] M. Hussain, *Automatic Recognition of Sign Language Gestures*, Master's Thesis, Jordan University of Science and Technology, 1999.
- [8] M. Handouyahia, D. Ziou, and S. Wang, "Sign Language Recognition Using Moment-based Size Functions", *In Proceedings International Conference on Vision Interface*, pp. 210–216, 1999.
- [9] S. Malassiotis and M. Strintzis, "Real-time Hand Posture Recognition Using Range Data", *Image and Vision Computing*, vol. 26, No. 7, pp. 1027–1037, 2008.
- [10] A. Licsar and T. Sziranyi, "Supervised Training Based Hand Gesture Recognition System", *In Proceedings of the International Conference on Pattern Recognition*, pp. 999–1002, 2002.
- [11] W. Freeman and M. Roth, "Orientation Histograms for Hand Gesture Recognition", *In Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, pp. 296–301, 1994.
- [12] H. Yoon, J. Soh, Y. J. Bae, H. S. Yang, "Hand Gesture Recognition Using Combined Features of Location, Angle and Velocity", *Journal of Pattern Recognition*, vol. 34, No. 7, pp. 1491–1501, 2001.
- [13] R. Klette, K. Schluns, A. Koschan, "Three-Dimensional Data from Images", *Computer Vision*, Springer, Singapore, 1998.
- [14] A. Al-Hamadi, O. Rashid, and B. Michaelis, "Posture Recognition using Combined Statistical and Geometrical Feature Vectors Based on SVM", *International Journal of Computational Intelligence*, vol. 6, No. 1, pp. 7–14, 2010.
- [15] L. Jin, C. Chen, L. Zhen, and J. Huang, "Real-Time Fingertip Detection from Cluttered Background for Vision-based HCI", *Journal of Communication and Computer*, vol. 2, No. 9, pp. —8, 2005.
- [16] J. Davis, G. Bradski, "Real-time Motion Template Gradients using Intel CVLib", *In Proceeding of IEEE ICCV Workshop on Framerate Vision*, 1999, 1–20. Year of Publication: 1999.
- [17] S. Khalid, U. Ilyas, S. Sarfaraz, A. Ajaz, "Bhattacharyya Coefficient in Correlation of Gary-Scale Objects", *Journal of Multimedia*, vol. 1, No. 1, pp. 56–61, 2006.
- [18] M. Elmezain, A. Al-Hamadi, B. Michaelis, "Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences", *The Journal of WSCG'08*, vol. 16, No. 1, pp. 65–72, 2008.
- [19] R. Niese, A. Al-Hamadi, B. Michaelis, "A Novel Method for 3D Face Detection and Normalization", *The Journal of Multimedia*, vol. 2, No. 5, pp. 1–12.
- [20] D. Comaniciu, V. Ramesh, P. Meer, "Kernel- Based Object Tracking", *IEEE Transactions PAMI*, vol. 25, No. 5, pp. 564–577, 2003.
- [21] J. Smith & A. Doe, "Gesture Recognition Using Polar Coordinates", *Journal of Machine Learning*, vol. 35, No. 2, pp. 134–150, 2020.
- [22] M. Brown & P. Wilson, "Noise Reduction in Motion Trajectories", *International Conference on Computer Vision*, vol. 22, No. 4, pp. 567–580, 2018.
- [23] L. Zhang & Y. Chen, "Hidden Markov Models for Gesture Classification", *Neural Computing and Applications*, vol. 33, No. 10, pp. 2234–2250, 2021.

- [24] I. V. Tetko, D. J. Livingstone, and A. I. Luik, "Neural Network Studies. Comparison of Over_Fitting and Overtraining", *Journal of Chemical Information and Computer Sciences*, vol. 35, No. 5, pp. 826–833, 1995.
- [25] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences", *Journal of WSCG*, vol. 16, No. 1, pp. 65–72, 2008.
- [26] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory", *In Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 519–522, 2008.
- [27] D. Comaniciu, S. Ramesh, and P. Meer, "Kernel-Based Object Tracking", *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 25, No. 5, pp. 564–577, 2003.

СИСТЕМА РОЗПІЗНАВАННЯ ЖЕСТІВ ДЛЯ УПРАВЛІННЯ ІОТ СИСТЕМАМИ

Максим Ферентц, Ігор Рабійчук, Андрій Фечан

Розвиток Інтернету речей (IoT) відкриває нові можливості для розробки інтелектуальних сервісів, що покращують взаємодію користувачів із навколишніми пристроями. Сучасні ІоТ-системи в основному використовують сенсорні екрани та мобільні додатки для керування, проте методи на основі жестів можуть значно розширити їхню функціональність. Запропоновано систему розпізнавання жестів, що застосовують для керування ІоТ-пристроями. Основою роботи системи є класифікація траєкторії руху пальців за допомогою прихованої марковської моделі (НММ). Система складається із трьох основних етапів: початкове виділення рук за допомогою кольорової та глибокої інформації, визначення кінчиків пальців на основі контурів руки, а також використання кластеризації в полярних координатах для екстракції динамічних характеристик. Алгоритми Баум-Велша та Вітербі застосовують відповідно для навчання та розпізнавання жестів. Результати експериментів показують, що розроблена система здатна класифікувати жести з урахуванням просторово-часової варіативності з високою точністю. Зокрема, середній рівень розпізнавання досягнув 98,61 % для навчального набору та 93,06 % для тестових даних. Запропонований підхід демонструє ефективність у складних умовах, включаючи зміни освітлення та часткове перекриття об'єктів у сцені.



Maksym Ferents is a 4th-year student specializing in Software Engineering at Lviv Polytechnic National University. His/Her research interests are focused on Internet of Things (IoT) systems, computer vision, and human-computer interaction (HCI), with a specific focus on developing innovative control methods.



Ihor Rabiichuk is a PhD student at the Institute of Computer Science and Information Technologies, Lviv Polytechnic National University, Ukraine.

His research focuses on swarm intelligence algorithms in dynamic environments, their application in adaptive and distributed systems, as systems, optimization methods well as the use of cellular automata for modeling and optimization processes.



Andriy Fechan D.Sc., Prof. Head of Department of Electronics and Information Technology of Lviv Polytechnic National University. Research interests cover optoelectronic devices based on organic materials and computer simulations of anisotropic liquids by molecular dynamics methods complex multi-component platforms.

Research results are published in more than 200 scientific papers and 25 patents by Ukraine.

Received: 16.07.2025, Accepted: 25.09.2025

ORCID: 0009-0006-0205-1233 (Ferents M.)

ORCID: 0000-0001-9970-5497 (Fechan A.)

ORCID: 0009-0002-0881-0406 (Rabiichuk I.)