2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

A Device Control System Using User-Defined Full-Body Gestures with HoloLens2

Yushin Mochizuki[1], Sarthak Pathak[2] and Kazunori Umeda[3]

Abstract— **This paper presents a novel system that allows users to operate devices through full-body gestures they define themselves. Gesture-based control systems are seeing widespread application for controlling a diverse range of devices. However, most existing systems rely on a predefined set of gestures. This fundamental limitation restricts not only the number of possible operations but also the overall system flexibility. The proposed system overcomes the limitation. The proposed system comprises two primary phases. In the initial definition phase, a user wearing a HoloLens2 defines a new gesture by directly manipulating the posture of a virtual avatar. This approach facilitates a highly visual and intuitive method for gesture creation. Throughout this definition process, an external camera system captures the user's movements and computes the skeletal joint data. Then, in the subsequent operation phase, the user can control devices by performing the defined gesture without needing to wear the HoloLens2. For recognition, the system relies solely on the external camera system. This results in a highly adaptable and extensible framework for gesture-based interaction.**

I. INTRODUCTION

In recent years, gesture recognition systems have been applied across a diverse range of fields. With the spread of the Internet of Things (IoT) and smart homes, gesture control is increasingly valued as an intuitive method for interacting with electronic devices. In medical and industrial settings, for instance, touchless interfaces are being developed to mitigate risks such as operational delays or contamination caused by physical contact with equipment[1]. Similarly, interfaces are being developed that enable users to intuitively control public displays through body movements.[2]. These varied applications highlight a growing demand for robust and user-friendly gesture recognition technologies.

Several studies have explored the use of gestures for controlling multiple devices. Yan et al., proposed a system where users control home appliances by performing a hand gesture within a "command space" associated with a specific command[3]. In our previous work, we proposed a method that utilizes the Microsoft HoloLens2[4] to visualize and adjust these command spaces, thereby providing a flexible control environment tailored to the individual user[5]. However, these existing approaches face significant limitations. The number of available operations is inherently constrained by the number of command spaces that can be established.

The objective of this paper is to develop a highly adaptable interface that empowers users to operate devices using user-defined full-body movements. To achieve this, we propose a system that leverages the HoloLens2 for the intuitive definition of these gestures. This paper primarily targets home and office environments, with the goal of enhancing usability and accessibility by providing a control environment that conforms to an individual's unique intuition and physical characteristics. Furthermore, the proposed method has the potential to be extended to scenarios requiring touchless operation, such as in healthcare, manufacturing, and public facilities.

II. RELATED WORKS

To create more effective gesture recognition systems, several studies have explored the concept of user-definable gestures. For example, Vogiatzidakis et al. developed a system for controlling multiple home devices with mid-air gestures, which allowed for significant spatial freedom[6]. However, a key limitation was that the gestures were restricted to a fixed set based on user surveys, and the system did not allow users to add their own gesture. Similarly, Ye et al. proposed an AR-based prototyping tool on smartphones for interaction design with IoT devices[7]. While this tool let users assign gestures to any location, the choice of gestures was limited to only three predefined types. These studies are supported by the findings of Nacenta et al., who reported that user-defined gestures are significantly easier for people to remember and feel more intuitive than system-defined ones[8].

The interaction between Mixed Reality (MR) and robotics is also an active area of research. Systems have been proposed that use HoloLens2 to operate a digital twin of a robot arm[9], and to interactively control wearable[10] and collaborative[11] robot arms through gestures. These studies show that MR devices can enhance operational freedom and usability.

Other research in gesture interaction includes a dynamic gesture recognition method for human-robot interaction[12] and gesture recognition for drone control[13]. While these studies demonstrate effectiveness for specific applications, they are designed for limited environments or purposes. A challenge remains in adapting them to more diverse operational needs and living environments. To address these limitations, this paper aims to build a general-purpose and flexible gesture control interface that users can freely define.

[1]Precision Engineering Course, Graduate School of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo, Japan. mochizuki@sensor.mech.chuo-u.ac.jp

[2]College of Engineering, Shibaura Institute of Technology, 3-7-5 Toyosu, Koto-ku, Tokyo, Japan. spathak@shibaura-it.ac.jp

[3]Department of Precision Mechanics, Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo, Japan. umeda@mech.chuo-u.ac.jp

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

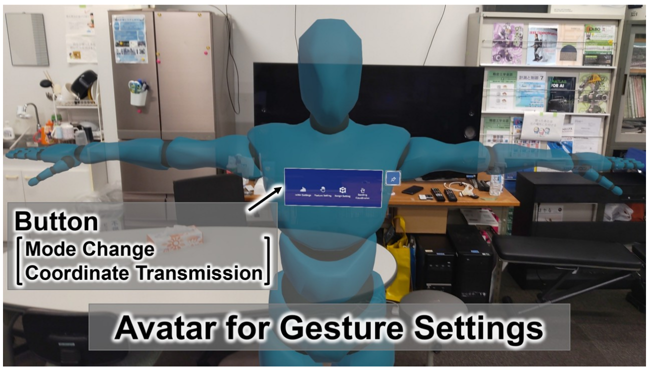


Fig. 1: System Overview

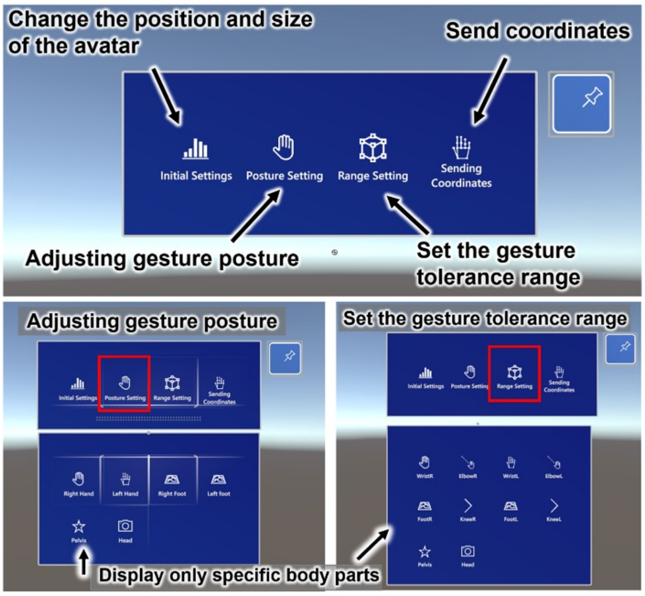

Change the position and size of the avatar
Send coordinates
Initial Settings
Posture Setting
Range Setting
Sending Coordinates
Adjusting gesture posture
Set the gesture tolerance range
Adjusting gesture posture
Set the gesture tolerance range
Right Hand
Left Hand
Right Foot
Left foot
Pelvis
Head
WristR
ElbowR
WristL
ElbowL
FootR
KneeR
FootL
KneeL
Pelvis
Head
Display only specific body parts

Fig. 2: The Role of Each Button

III. PROPOSED SYSTEM

A. System Overview

Our system consists of two main phases: a definition phase and a recognition phase.

In the definition phase, the user wears a HoloLens2 to define a new gesture. First, a system using an external camera captures the user's posture, and this posture data is sent to the HoloLens2. This data sets the initial posture of a virtual avatar. The user then refines the avatar's posture and specifies the gesture's tolerance range. The defined gesture information is then sent to the gesture recognition module and saved.

In the recognition phase, the system matches the user's current movements captured by the external cameras with the saved gesture definitions. Upon finding a match, the corresponding device operation is executed. Because this recognition process use the saved data, the user can perform gestures without wearing the HoloLens2.

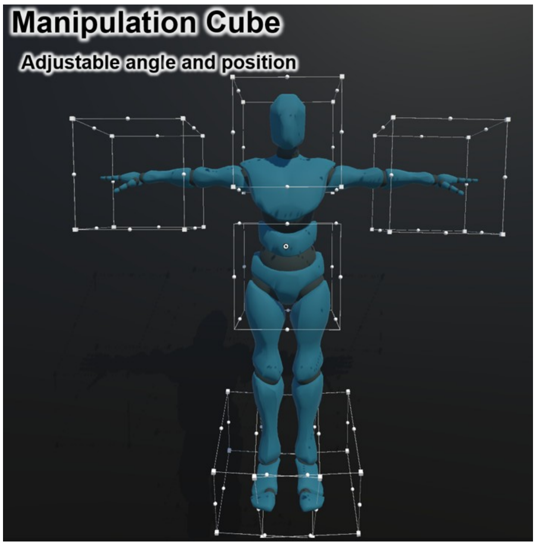


Fig. 3: Manipulation Mode

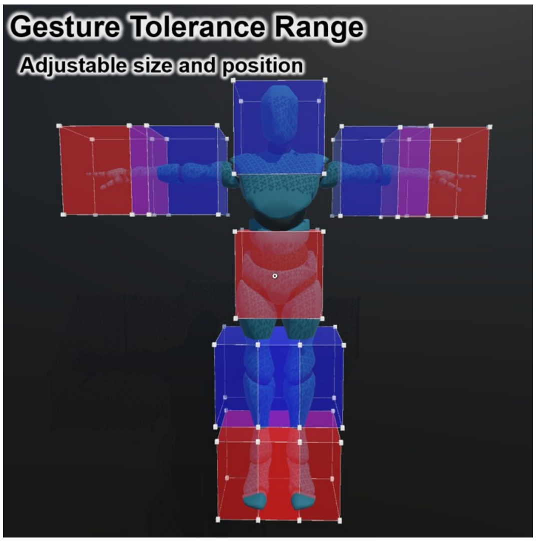


Fig. 4: Gesture Tolerance Range

B. System Environment

The gesture definition module runs on the HoloLens2, which renders the avatar and allows the user to perform the definition task. The gesture recognition module, as well as the initial avatar posture setting during the definition phase, are managed by the external camera system.

In principle, our method can operate in any environment where 3D coordinates can be obtained using a minimum of two cameras or one RGB-D camera. For this study, we constructed an environment with four CCD cameras installed at the ceiling corners of a room. This setup ensures stable recognition regardless of the user's position or orientation.

We apply OpenPose[14] to the captured images to estimate the 2D skeletal joints of the human body in real-time. By applying stereo vision to the information from these multiple viewpoints, we reconstruct the 3D coordinates of each skeletal joint.

C. Gesture Definition Interface

In our system, the user wears the HoloLens2 to see a life-sized avatar from a third-person perspective, as shown in Fig.1[15]. The user defines a gesture by directly manipulating this avatar. The avatar is equipped with an Inverse Kinematics (IK) function, ensuring that adjustments result in natural and physically plausible joint movements. This feature facilitates the creation of realistic postures and prevents poses that would exceed the natural range of human motion.

The configuration process is mode-based, allowing for intuitive operation. The user can switch between modes by pressing a series of buttons, shown in Fig.2, from left to right.

1) Automatic Posture Synchronization: Manually adjusting the avatar's posture from an initial default state can be a significant burden for the user. To reduce this burden, our system includes a function that automatically synchronizes the avatar's posture with the user's current posture at the beginning of the definition process.

This process is executed as follows. First, the external camera system acquires the user's skeletal information and calculates the 3D relative coordinates of major joints (wrists, ankles, head, and waist) with the chest as the reference point. Next, this coordinate data is continuously collected for 5s, and the time-average is calculated. The resulting average coordinate data is then sent to the HoloLens2 and applied to the corresponding parts of the avatar. This enables the user to start the gesture definition process more efficiently from a pose that closely matches their own gesture.

2) Gesture Definition using HoloLens2: After the automatic adjustment, the user performs more detailed manual adjustments on the HoloLens2. First, the user adjusts the overall scale and spatial position of the avatar. The avatar's scale is matched to the user's height so that the user can intuitively check the posture by overlaying their own body with the avatar after the definition is complete.

Next, the user defines the target posture. As shown in Fig.3, manipulation cubes are displayed on the avatar's major joints (wrists, ankles, head, and waist). The user can drag these manipulation cubes to precisely set the desired gesture posture. Furthermore, by using the joint selection buttons shown in Fig.2, the user can display only the manipulation cubes for specific joints, allowing for more focused adjustments.

The user then sets the tolerance range for the gesture posture, as shown in Fig.4. This setting accommodates for errors in reproducibility and slight inconsistencies in how the user performs the arm movements. By adjusting the shape of this tolerance range according to the user's intent, the system can prevent misrecognition between similar movements while enabling flexible and intuitive operation. Specifically, tolerance boxes corresponding to the positions of the wrists, elbows, knees, ankles, waist, and head are displayed on the avatar. A gesture is recognized when all specified joints are within their corresponding tolerance boxes. By adjusting the position and size of these tolerance boxes, the user can flexibly define the spatial tolerance for the target movement. In this setting as well, the buttons shown in Fig.2 are displayed, allowing the user to manipulate the tolerance box for a specific body part.

After all definitions are complete, pressing the save button sends the defined gesture information to the gesture recognition module, where the gesture information is stored.

D. Gesture Coordinate Transmission

Once a gesture is defined by the user on the HoloLens2, the relevant information is transmitted to the gesture recognition module over a network connection. For this data transfer, we employ the TCP/IP protocol, which ensures efficient transmission of the lightweight coordinate data used by our system.

The transmitted data package primarily consists of the vertex coordinates for the tolerance boxes that constitute the gesture's tolerance range. All coordinates are defined relative to the avatar's chest, establishing a local coordinate frame for the gesture. This data effectively outlines the spatial regions within which the user's joints must be positioned for a successful match. The gesture recognition module then uses this information to perform its matching recognition.

This clear separation of the definition and recognition phases is a key architectural feature of our system. It enables users to control devices via gestures without needing the HoloLens2, following a one-time initial setup.

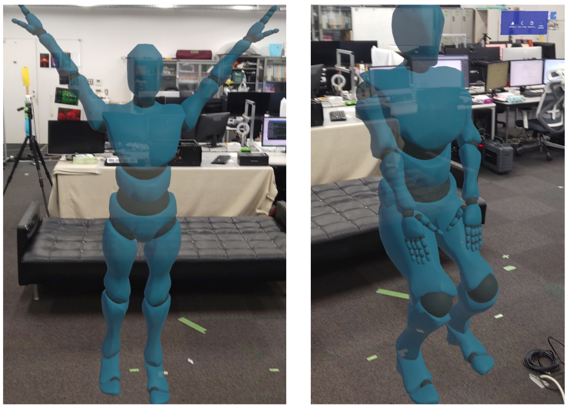
E. Gesture Recognition Method

The gesture recognition module operates based on the user's skeletal information, which is acquired in real-time from the external camera system. It calculates the 3D coordinates of the wrists, elbows, knees, ankles, waist, and head for use in gesture recognition. To prevent misrecognition from unintentional movements, the system begins the gesture recognition only after detecting that the user's body has been static for a certain period.

For the recognition, the system first establishes a relative coordinate system fixed to the user's body to account for differences in height and position. The chest coordinate is set as the origin, the line connecting the shoulders as the Y-axis, the body's height direction as the Z-axis, and their cross product as the X-axis. All target joint coordinates are then transformed into this relative coordinate system. This process reduces the influence of individual differences and enables stable recognition.

To ensure consistency during a sustained posture, the orientation of this coordinate system remains fixed while the user is static. The origin, however, continuously tracks the chest's movement. This dynamic origin allows the system to correctly recognize gestures even if the user changes their overall position, such as by transitioning from standing to sitting.

Ultimately, a gesture is deemed successful if all specified target joints, once transformed, are located within their respective, user-defined tolerance boxes. Through this comprehensive recognition method, our system achieves stable and high-precision performance for a wide range of gestures.



(a) Y-Pose (b) Sitting Pose

Fig. 5: Gestures Defined in Experiment 1

IV. EXPERIMENT

A. Experiment Overview

To evaluate the effectiveness of our proposed system, we conducted two experiments with 12 male and female participants in their 20s. We assessed the system based on three key metrics: operation time, task accuracy, and user experience, the last of which was measured via subjective workload and usability questionnaires.
(This experiment was approved by the ethics committee at Chuo University.)

1) Experiment 1: Experiment 1 aimed to compare the effects of predefined versus user-defined gestures on task performance and subjective workload. Participants performed two common gestures shown in Fig.5, a "Y-pose" and a "sitting pose", under two different conditions: a "predefined condition," where they used gestures set by an experimenter, and a "user-defined condition," where they defined and used their own gestures with our system. For the predefined condition, the experimenter set gestures to match common motion images, with a uniform tolerance range of a 35 cm cube for all joints. We measured task completion time and accuracy as quantitative data. Task completion time includes the time the user performs gestures.

Following the tasks, subjective workload was assessed using the NASA-Taylor Load Index (NASA-TLX)[16]. This standard method evaluates perceived workload across six subscales: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. An overall workload score is derived from a weighted average of these scales. Crucially, this evaluation focused solely on the workload associated with performing the gesture-based control task.

2) Experiment 2: The purpose of Experiment 2 was to evaluate the overall usability of the entire process, from defining a gesture to performing a task. Participants were asked to define an arbitrary gesture using our system and then immediately use it to complete a specific task. We measured task completion time and accuracy.

TABLE I: Experiment 1 Results

	Mean Time to Detect	Percentage not Recognised
Y Pose (Predefined condition)	2.71 s	30 %
Y Pose (User-defined)	2.75 s	3 %
Sitting Pose (Predefined condition)	3.60 s	61 %
Sitting Pose(User-defined)	4.32 s	42 %

Additionally, we evaluated the overall usability of the system using the System Usability Scale (SUS)[17]. The SUS is a questionnaire that quantitatively assesses system usability through the following 10 items:
1: I think that I would like to use this system frequently.
2: I found the system unnecessarily complex.
3: I thought the system was easy to use.
4: I think that I would need the support of a technical person to be able to use this system.
5: I found the various functions in this system were well integrated.
6: I thought there was too much inconsistency in this system.
7: I would imagine that most people would learn to use this system very quickly.
8: I found the system very cumbersome to use.
9: I felt very confident using the system.
10: I needed to learn a lot of things before I could get going with this system.

Participants responded to each item on a 5-point Likert scale from "Strongly Agree" to "Strongly Disagree." The SUS score is calculated by summing the scores for odd-numbered questions, subtracting 5, and adding this to 25 minus the sum of scores for even-numbered questions, then multiplying the result by 2.5. This experiment evaluated the entire process, from gesture definition with HoloLens2 to recognition.

B. Experimental Conditions

In each gesture task, participants performed three trials per gesture. The gesture recognition phase for all experiments was conducted without wearing the HoloLens2.

For the evaluation, we measured operation time and accuracy. Operation time was measured from the start signal of a task until the system correctly recognized the gesture. Accuracy was the success rate over the three trials. A trial was considered a failure if the gesture was not recognized within 10s from the start.

C. Experiment 1: Results and Discussion

1) Results: The results for operation time and failure rate in Experiment 1 are shown in Table I. For the Y-pose, the average operation time in the predefined condition was 2.71s with a 30% failure rate, whereas in the user-defined condition, the average time was 2.75s with a 3% failure rate. Similarly, for the sitting pose, the predefined condition resulted in an average operation time of 3.60s and a 61% failure rate, compared to 4.32s and a 42% failure rate in the user-defined condition. These results confirm that while operation time slightly increased, the failure rate for

TABLE II: Average NASA-TLX Score

	Predefined condition	User-defined
Mental Demand	22.1	17.8
Physical Demand	14.1	9.9
Temporal Demand	11.3	7.5
Effort	26.3	20.8
Frustration	14.0	8.6
Performance Demand	51.3	29.4
Weighted Workload	25.9	18.0

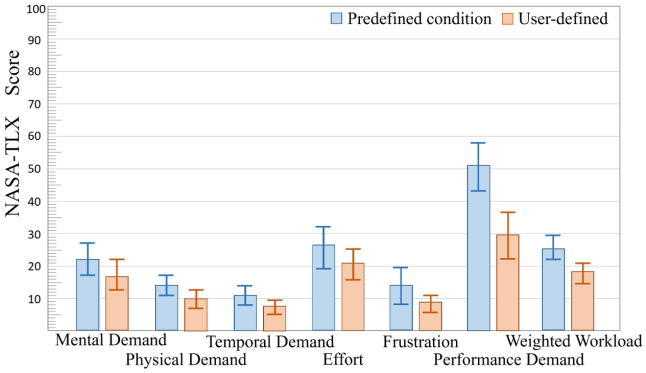


Fig. 6: NASA-TLX Score

both gestures improved significantly when users defined the gestures themselves.

The NASA-TLX results are presented in Table II and Fig.6. For all six subscales, the average scores for the user-defined condition were lower than those for the predefined condition. The overall workload score was also significantly lower for the user-defined condition (M = 18.04, SD = 10.11) compared to the predefined condition (M = 25.88, SD = 11.70). Before applying the test, we visually inspected the distribution of the difference scores and confirmed no significant deviation from normality. A paired-samples t-test revealed that this difference was statistically significant (t(11)=3.71, p=0.003). This indicates that using user-defined gestures significantly reduces subjective workload.

2) Discussion: Overall, the results indicate that user-defined gestures yielded a higher success rate and lower subjective workload, although they came at the cost of a minor increase in operation time. We attribute this increased operation time to a characteristic of our current definition interface's recognition logic, which requires all target joints to be strictly within their defined tolerance boxes. Consequently, if a participant defined a very narrow tolerance range, even slight deviations during performance could lead to recognition delays. Future work could address this by introducing features to assist with tolerance setting, or by implementing a more flexible recognition algorithm that can accommodate a certain degree of error.

Furthermore, the sitting pose had a generally higher failure rate than the Y-pose. Reproducing the Y-pose primarily requires conscious positioning of the arms, a relatively simple task. The sitting pose, in contrast, demands accurate positioning of the entire body, including the less-consciously controlled legs and waist. This is likely because the increased complexity imposes a higher cognitive load on the user, leading to greater performance variability and, thus, a higher failure rate.

TABLE III: Experiment 2 Results

	Mean Time to Detect	Percentage not Recognised
Freely Pose	2.77 s	17 %

D. Experiment 2: Results and Discussion

1) Results: The results for operation accuracy and time with freely defined gestures in Experiment 2 are shown in Table III. The average operation time was 2.77s, and the average failure rate was 17%. The usability evaluation using SUS resulted in an average score of 64.0, which is below the general average of 68.1[18].

2) Discussion: In Experiment 2, the gestures defined by participants ranged from simple Y-poses to more complex ones. The ability to keep the average failure rate at 17% for such a wide variety of gestures suggests the high degree of freedom and versatility of our system.

However, the SUS score being below average indicates that there are usability issues. In particular, low ratings were prominent for item 2, “I found the system unnecessarily complex,” and item 4, “I think that I would need the support of a technical person to be able to use this system.” We attribute this perceived complexity to two primary factors. First, the large number of manipulable body parts and parameters can be overwhelming for new users. Second, it can be difficult to intuitively predict how adjustments to the avatar's posture and tolerance boxes will impact the final recognition performance.

Therefore, future improvements will focus directly on enhancing usability. Potential directions include simplifying the interface by reducing the number of manipulation targets and developing a function that automatically suggests an appropriate tolerance range based on the defined posture.

V. CONCLUSIONS

In this paper, we have developed a novel interface that empowers users to contactlessly operate devices using their own user-defined gestures. We demonstrated that the HoloLens2 serves as a visual and intuitive medium for defining not only a gesture's posture but also its precise tolerance ranges. A key feature of our architecture is that following this initial definition phase, device control is achieved through gestures alone, eliminating the need for the user to wear an HoloLens2.

Our evaluation experiments demonstrated the system's capacity to recognize a diverse range of user-defined gestures. Furthermore, the results confirmed that using these personalized gestures reduces the subjective workload associated with the control task. However, our findings also highlighted clear usability challenges within the current gesture definition process, pointing to a key area for future enhancement.

For future work, we plan to improve usability by introducing few-shot learning for image-based recognition to supplement the primary method and a dynamic threshold adjustment algorithm that adapts to user characteristics. These enhancements aim to simplify the definition process and further stabilize recognition accuracy. Furthermore, while the current system only recognizes static postures, we plan to incorporate a global coordinate system in addition to the body-relative one. This will enable the recognition of dynamic gestures that consider the user's position and orientation, thereby enabling more complex interactions.

REFERENCES

[1] M. G. Jacob, J. P. Wachs, and R. A. Packer, "Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images,“ Journal of the American Medical Informatics Association, vol.20, no.1, pp.101–107, Jan. 2013.

[2] Q. Wang and Z. Xie, "ARIAS: An AR-based interactive advertising system“, PLoS ONE, vol.18, no.1, e0280000, 2023.

[3] S. Yan, Y. Ji, and K. Umeda, "A System for Operating Home Appliances with Hand Positioning in a User-definable Command Space," 2020 IEEE/SICE International Symposium on System Integration (SII), pp.366-370, 2020.

[4] Microsoft Corporation. (2024). Microsoft HoloLens2. Accessed: Aug. 13, 2025. [Online]. Available: https://www.microsoft.com/ja-jp/hololens

[5] Y. Mochizuki, M. Yokota, S. Pathak, and K. Umeda, "Visualisable and Adjustable Command Spaces for Gesture-based Home Appliance Operation System via HoloLens2,“ 2025 IEEE/SICE International Symposium on System Integration (SII), pp. 1405-1410, 2025.

[6] P. Vogiatzidakis and P. Koutsabasis, "Mid-Air Gesture Control of Multiple Home Devices in Spatial Augmented Reality Prototype“ Multimodal Technologies and Interaction(MTI), vol. 4, no. 3, pp. 61 , 2020.

[7] H. Ye and H. Fu, "ProGesAR: Mobile AR Prototyping for Proxemic and Gestural Interactions with Real-world IoT Enhanced Spaces“, CHI '22: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, No. 130, pp. 1–14, 2022.

[8] M. A. Nacenta, Y. Kamber, Y. Qiang, and P. O. Kristensson, "Memorability of Pre-designed and User-defined Gesture Sets“ Proceedings of the 31st Annual ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2013), pp. 1099–1108, 2013.

[9] A. Smith and M. Kennedy, "An Augmented Reality Interface for Teleoperating Robot Manipulators“, arXiv preprint arXiv:2402.18928, 2024.

[10] H. Jing et al., "Human Operation Augmentation through Wearable Robotic Limb Integrated with Mixed Reality Device“, Biomimetics, vol. 8, no. 7, p.531, Nov.2023.

[11] A. Pittiglio and X. Yang, "Augmented Human-Robot Collaborative Bending. Human-robot collaboration for biogenic material bending-active assembliess“, in Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), vol. 1, pp. 621-630, 2024.

[12] Q. Gao et al., "Dynamic Hand Gesture Recognition Based on 3D Hand Pose Estimation for Human–Robot Interaction,“ IEEE Sensors Journal, vol. 21, no. 15, pp. 16476–16486, 2021.

[13] M. Iskandar, K. Bingi, R. Ibrahim, M. Omar, and P. A. M. Devan, "Hybrid Face and Eye Gesture Tracking Algorithm for Tello EDU RoboMaster TT Quadrotor Drone,“ 2023 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1-6, 2023.

[14] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,“ in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172-186, 2021.

[15] Adobe, "Y Bot," Mixamo. Accessed: Aug. 15, 2025. [3D model]. Available: https://www.mixamo.com

[16] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research“, Human Mental Workload, pp. 139-183, 1988.

[17] J. Brooke, "SUS–a quick and dirty usability scale," Usability Evaluation in Industry, pp. 189-194, 1996.

[18] J. Sauro, "A Practical Guide to the System Usability Scale," Measuring Usability LLC, 2011.