

Technical Paper

Experimental analysis of augmented reality interfaces for robot programming by demonstration in manufacturing

Chih-Hsing Chu^{*}, Chen-Yu Weng

Department of Industrial Engineering & Engineering Management, National Tsing Hua University, Hsinchu, Taiwan

ARTICLE INFO

Keywords:

Augmented reality
Robot
Programming by demonstration
Eye gaze
Head gaze
Hand ray, user interface

ABSTRACT

Augmented Reality (AR) technology has been effectively utilized to support various manual operations in the manufacturing industry. An important application is serving as a user interface for human-robot collaboration. This paper presents an experimental study on the feasibility of robot programming by demonstration (PbD) through AR interfaces in the context of manufacturing. Our focus is on comparing the pointing and line tracing processes using three input methods provided by an AR headset: hand ray, head gaze, and eye gaze, based on both objective and subjective measures obtained from the experiment. The hand ray method performs the best in terms of accuracy, precision, and completion time in most experimental conditions. The SUS and NASA-TLX scores indicate acceptable usability for the hand ray method but low usability for the others. A prototyping AR tool using the hand ray as non-contact input in the real world is developed for motion planning of an industrial robotic arm. A test case of tire mold welding verifies the feasibility of the AR tool while also showing its limited capability in precision manufacturing. This work demonstrates a new approach for robot PbD on tangible objects enabled by AR.

1. Introduction

With the rapid advancement of information and communication technology, Industry 4.0 has emerged as an effective approach to realize product mass customization. Augmented Reality (AR) is recognized as one of the nine enabling elements that help implement the concept of Industry 4.0. Indeed, AR applications have been deployed to enhance the capabilities and work efficiency of manual operations across various industries [1]. Particularly, the manufacturing industry is expected to have a growing demand for these assistive tools in the near future. They have been effectively introduced to a wide range of manufacturing tasks, such as assembly/disassembly, maintenance, inspection, production simulation, and on-site personnel training [2]. On the other hand, industrial robots are an important automation technology that no longer exists stand-alone in most modern factories. Human-robot collaboration (HRC) becomes a common practice that combines high accuracy, strength, and repeatability of robots with high flexibility and adaptability of humans for optimal overall productivity [3]. This requires an interactive interface between humans and robots for effective communication and facilitating collaborations. Previous studies have confirmed that AR successfully serves as such a role in manufacturing applications

[4–6].

Fully autonomous robot operations frequently become impossible or impractical to accomplish in complex manufacturing tasks on the shop floor [7]. Under such circumstances, robot motion planning by human is a more feasible approach to reliably complete the tasks. Different approaches serve this purpose in the current industrial settings. First, robot motion planning software, which has been existing for decades, calculates robot motions offline through simulation that satisfy movement constraints, avoid collisions, and possibly optimize some aspect of the movement [8]. However, these features typically operate on 3D models of the work environment and the robot, which can be time-consuming to create or simply unavailable on-site. Robot Programming by Demonstration (PbD) technology has been developed to overcome this problem. In the typical PbD procedure, a human planner manipulates a robot in an actual manufacturing environment by using a teaching pendant or a similar instructive device. The detailed motions thus generated subsequently drive the robot movements while executing its actual task. The demonstration procedure is often tedious, error-prone, and not intuitive for the planner when maneuvering a robot in 3D space. Cobot (or collaborative robot) has been recently developed to facilitate direct human-robot interaction within a shared space. A human can conduct

^{*} Corresponding author.

E-mail address: chchu@ie.nthu.edu.tw (C.-H. Chu).

<https://doi.org/10.1016/j.jmansys.2024.03.016>

Received 1 October 2023; Received in revised form 13 February 2024; Accepted 30 March 2024

Available online 18 April 2024

0278-6125/© 2024 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

motion planning by directly grasping and directing a cobot, but the process is still time-consuming. More restrictively, the cobot design is not intended for performing heavy-duty tasks, which limits its applicability in manufacturing processes.

These shortcomings have inspired extensive research on designing new user interfaces and interaction methods for robot programming. Previous studies have shown the practicality of AR as an intuitive interface in PbD that facilitates interactions among humans, robots, and the environment by superimposing instructive information onto the real world [9–11]. Robot motion planning based on AR technology has been employed across a wide spectrum of manufacturing applications, including pick-and-place [9], welding [10], spraying [12,13], and machining [14]. It is advantageous to evaluate whether the motion planning results created through AR interfaces, particularly those implemented using a head-mounted display (HMD) device, can meet the actual manufacturing requirements in terms of positional precision. Moreover, AR interface designs, especially with various input methods, may impact the effectiveness of robot motion planning. To answer these questions, this research conducted an experimental study on robot PbD using AR interfaces to characterize its capability from manufacturing perspectives. In the study, subjects performed pointing and line tracing in 3D space using three input methods: hand ray, head gaze, and eye gaze provided by an AR headset under varied experimental conditions. Evaluation criteria of the task performance including accuracy, precision, and completion time were collected from the experiment to cross-compare the three methods. The SUS and NASA-TLX scores obtained also provide insights into individual usability. A prototyping AR tool adopting the optimal input interface was implemented for motion planning of an industrial robotic arm that performs welding in tire mold repair. The test results have verified the feasibility of robot motion planning on real work parts assisted by non-contact AR interfaces. This work highlights the potential and limitations of current AR technology in human-robot collaboration (HRC) for manufacturing processes.

2. Related studies

This section reviews the previous studies related to AR assisted robot PbD in manufacturing. The focus is on how AR technology has been applied to improve human situation awareness in robot motion planning. Various interaction devices are available for displaying information and interacting with users in AR, including head mounted display (HMD), hand-held display (HHD), monitor, and projector [2]. Zaeh and Vogl [14] proposed an AR-based method for intuitive and efficient programming of industrial robots. In this method, motion trajectories and target coordinates are interactively visualized and manipulated in the work environment using laser projection. Experimental results showed that the method significantly reduces the robot programming time and achieves a positioning precision of 0.5 mm for target points. Fan et al. [15] developed an AR-based approach for planning the motion path and orientation of the end-effector for an industrial robot. Users can interactively specify a curve path by giving a list of control points, define the orientation of the end-effector associated with each control point, and generate a ruled surface representing the path through a display monitor. Veiga et al. [16] implemented a monitor-based AR-assisted application for a robotic workcell to create ceramic tableware. Evaluation tests demonstrated that this application outperforms offline software solutions and teach pendant programming in terms of the time required for programming. Similar techniques that overlay virtual information onto real scenes to assist robot planning using computer monitors can also be found in the past studies [17–19]. The information displayed using projectors and monitors can be easily blocked by other objects existing in the real environment. AR applications deployed on these devices often remain fixed in location, thus lacking flexibility in use.

Handheld devices such as smart phones and tablets have also been used for AR-based robot programming [20–25]. Chacko and Kapila [22]

presented a mobile AR interface aimed at facilitating human-robot interaction for pick-and-place tasks in a shared working environment. The AR interface was designed to reduce development costs and effort. De Pace et al. [23] evaluated an AR handheld interface to control a robotic manipulator for tasks involving the creation of robot paths in 3D space. The evaluation results suggested that these types of interfaces work moderately well for controlling the manipulators, indicating that there is still room for improvement and further research. Hügler et al. [24] developed a mobile AR application based on the concept of hybrid robot programming, which combines manual haptic guidance of the end-effector with programming approaches using non-haptic pointing gestures for spatially defining poses and trajectories. An evaluation user study indicated that the hybrid programming method outperforms traditional teaching pad and CAD-based offline programming approaches. Kapinus et al. [25] developed an AR robot programming interface based on a mobile device to meet the requirements of programming robots for low and medium complexity tasks in a shared collaborative environment. The assessment results from a simple experiment validated the usability of the programming interface. Users can move freely in a manufacturing environment, but using a smartphone or tablet requires at least one hand, which limits their interaction with physical objects and their ability to perform manual operations [2]. HMD is the most common device deployed in AR applications in smart manufacturing, so the following literatures will mainly focus on robot programming based on this device.

2.1. AR-based robot programming

Ong et al. [11] proposed an AR assisted robot programming system (ARRPS) that provided faster and more intuitive programming processes than conventional methods. With an AR headset and a handheld pointer for interaction, users could freely move around a workplace to define 3D points and paths for the real robot to follow. Algorithms with real-time sensor data were applied for robot motion planning, collision detection, and plan validation. A prototyping system of AARPS was tested in two applications, welding and pick-and-place operations, to validate the functionality of the system. The results of a questionnaire based on the feedback from twenty participants confirmed that the AR interface was user-friendly and intuitive. Ostanin and Klimchik [26] developed a Mixed Reality (MR) based system for interactive programming of industrial robots, which consisted of main functions such as geometric path planning, kinematics analysis, optimal trajectory planning, and simulation. The system allowed users wearing Microsoft HoloLens to give operation commands to the robot manipulator without the need of previous programming experiences. Feedback from a group of experts was collected from three test motions: pick and place, circular, and rectangular trajectories, to demonstrate the usability of the MR-based system. Puljiz and Hein [27] presented an AR-based system mainly implemented by using open-source tools that allowed planning of end-to-end motion in human-robot interaction. New generation of AR HMDs provided technical features that enabled a flexible and collaborative robot work cell, containing various aspects from setting up the working environment, through programming, collision avoidance, learning, and finally interaction with the programmed robot. Yigitbas et al. [28] simplified the process and complexity of robot programming by combining AR with principles of end-user development. A prototyping AR-assisted tool projected a robot model as well as a programming environment onto the target working space through an HMD. Usability evaluation conducted by domain experts confirmed the potential of the tool to ease the current robot programming practice. Solanes et al. [29] proposed an AR-based interface for robot teleoperation, aiming to replace traditional teaching pendants and enable users to perform remote tasks. A two-stage experiment using a 6 R industrial robot manipulator was conducted to validate the AR interface. First, a teleoperation experiment on conditioning a car hood surface demonstrated the functionalities and performance of the proposed approach. It

was assumed that the target locations of the car hood were previously provided by other systems. Next, usability tests showed that the proposed interface was more intuitive, ergonomic, and easy to use. The velocity of the teleoperation task significantly increased, regardless of the user's previous experience in robotics or AR.

2.2. User interfaces in AR-based robot programming

Lambrecht and Krüger [30] introduced a spatial programming system for industrial robots that utilized a handheld device and a motion tracking device. The system provided functional modules for hand-gesture based definition of poses, trajectories, and tasks in robot programming. Araiza-Illan et al. [31] proposed an AR-based system to re-program robot packing through simple hand gestures and information collected by a HoloLens. Users specified an object to its corresponding storage area using hand gestures through the AR interface. An industrial robot then executed the pick and place task of the object following the motion commands thus generated. A similar hand-gesture interface enabled by HoloLens was used in robot programming for pick-and-place motion [9,32].

Eye gaze has been used in HRC as an intuitive form of human interaction that provides rich information about user's mental state and intent. Yan et al. [33] developed an AR headset that integrated foveated vision detection and eye tracking to reduce cognitive load in repetitive tasks in a warehouse. In a usability test with 33 participants performing parcel scanning tasks using the headset, the AR interface consistently maintained high scanning efficiency and reduced cognitive load across tasks of varying difficulties. Chan et al. [34] designed and implemented a joint action framework for human-robot collaboration. This framework utilized AR technology and user eye gaze in HoloLens to enable bidirectional communication of intent. A user study involving simple block placement by a robot showed that the framework improved the task efficiency, trust, and fluency.

Head-based interaction, namely head gaze or head pose, has been actively studied in the areas of 3D user interface, VR/AR [35], and wearable computing [36]. Head-pose-based pointing has proven to be an effective method for interacting with virtual objects in commercial headsets like Oculus Rift and Microsoft HoloLens, eliminating the need for hands or hand-held pointing devices. Arevalo Arboleda et al. [37] combined hands-free multimodal interaction methods, including a head-gaze-based cursor for pointing and speech commands to trigger actions, in remote robot manipulation and grasping. An evaluation experiment of pick-and-place tasks verified improvements in user performance compared to a baseline condition.

Previous studies have also adopted multi-modal AR interfaces in robot programming, particularly to reduce the mental workload associated with the user [38–40]. Not all task-related information is most effectively communicated visually and excessive visualization information can often cause perceptual overload for users. For this purpose, Chan et al. [38] developed an AR user interface that integrated gestural control and haptic feedback for programming and controlling a robotic arm. Liu et al. [39] introduced the robot PbD system, InstruMentAR, which features a multimodal interface incorporating gestural input, haptic feedback facilitated by a hand-worn pressure sensor, and voice recognition. This system was designed to capture the user's step-by-step manipulations on virtual control panels and automate the process of tutorial authoring. Sita et al. [40] implemented a control system for an industrial manipulator through user interactions with MR content displayed in the Microsoft HoloLens. The manipulator can operate in two modes: either through point-to-point instructions given by the user using the HMD, or direct manipulation of a target that the robot will track in 3D space.

2.3. Comparative studies on AR interfaces for robot programming

Previous research compared different pointing techniques in AR,

including head pose, finger directing, and eye gaze, using 3D projection [41], stereoscopic display [42], and desktop screen [43], respectively. The deployment of these devices encounters difficulties on the shop floor, limiting their use in AR applications for modern manufacturing [2]. Fewer studies have experimentally evaluated task performances in AR interfaces developed for HMDs, especially from an application perspective. Kytö et al. [44] conducted an experimental study on head and eye gaze pointing techniques in wearable AR. Their focus was to determine the merits of each method by comparing speed and pointing accuracy. The experimental results showed that eye gaze input was faster, but head pointing allowed greater targeting accuracy. The implications of these results were demonstrated through two implementations for precise menu selection and online improvement of gaze calibration in AR interfaces. Fiducial markers are commonly used to integrate the virtual space with the real world in an AR scene. Blanke-meyer et al. [9] estimated the marker tracking accuracy and precision from different distances and angles relative to a HoloLens. The estimation results indicated an accuracy of 1–2 mm and a precision of 3–5 mm were reachable, meeting the requirements in a human-robot collaborative assembly workplace. However, they did not consider potential deviations induced by input interface or human operation. On the other hand, very few studies have examined the performance of trajectory tracing with wearable AR platforms. Condino et al. [45] experimentally analyzed a proprietary AR headset designed for guiding complex 3D trajectory tracing tasks. The quantitative evaluation results indicated that over 94 % of traced trajectories remained within a 1 mm error margin relative to the target on a 3D-printed replica of a planar structure. Different from the robot PbD scenario, the subjects directly manipulated a desktop robot tip in the tracking task with instant haptic cues while receiving simultaneous visual cues from the headset. Krupke et al. [46] compared two different multimodal human-robot interaction techniques for selecting a location on a target object using head orientation and hand-ray pointing, both in combination with speech commands. The experimental results showed that the heading-based interaction is more precise, efficient, and involves less workload in MR-based pick-and-place scenarios.

While previous studies have confirmed the effectiveness of AR as an interactive interface for assisting robot PbD in manufacturing tasks, the literature review above shows several areas for further research. Most studies focused on interface design, system development, and usability evaluation compared to traditional robot programming methods. More evidence is needed to determine if robot PbD through AR interface is suitable for precision tasks and to characterize the technical capabilities of the current technology from the application perspectives. To address this need, this research conducts an experimental study on AR interfaces in robot PbD using a commercial HMD with various input methods. The focus is to evaluate the work performances of the AR interfaces in two common tasks of robot programming, pointing and profile tracing, using both objective and subjective measures. The experimental findings may offer insights into the current state of AR-assisted PbD from a practical perspective of manufacturing. A prototyping AR tool was implemented for a test case of tire mold welding to demonstrate the feasibility of robot motion planning on real objects using non-contact input approaches. The remainder of this paper is organized as follows. Section 3 describes the experiment design of the pointing and tracing tasks. Section 4 shows the analysis results of the experimental data, cross-compares various input methods, and suggests possible explanations for their differences. Section 5 highlights the prototyping tool and the robot welding process of tire molds through AR interfaces. The last section presents the concluding remarks and suggestions for future research.

3. Experiment design

This section describes the details of the evaluation experiment. Subjects wearing a Microsoft HoloLens 2 complete the pointing and tracing tasks using AR interfaces with three input methods under



Fig. 1. A subject is performing a task in the experimental.

Table 1
The experimental factor settings.

Factor	Settings
Task	Pointing, Line Tracing
Distance (cm)	(d1, d2) = (80, 50), (130, 100)
Input Method	Hand Ray, Head Gaze, Eye Gaze

different conditions. Objective measures of their work performance include accuracy, precision, and completion time for each task. These assessment criteria are closely related to both the manufacturing quality and efficiency of robotic welding programming assisted by AR. Moreover, subjective measures are collected using the NASA-TLX and SUS questionnaires after the experiment. They compare the workload and usability induced by the three input methods through the AR user interface for robot planning. The motion of an industrial welding robot is primarily defined by basic geometric elements such as points, lines, and arcs specified on planes. Similar to most previous studies, this work only compares the work performances of pointing and line tracing using different input methods through AR. The distance varied in the experiment was determined considering that the human operator is normally located within a similar range to a work part in robotic welding.

The following research questions will be explored based on the measures obtained from the experiment.

- Do the three input methods result in different performance outcomes in AR-assisted pointing and line tracing?
- Do the three input methods result in different workload perceived by users in AR-assisted pointing and line tracing?
- How does the target distance influence the performance outcomes in AR-assisted pointing and line tracing?
- To quantitatively evaluate performance outcomes in AR-assisted pointing and line tracing to determine if they meet the precision requirements of robot programming in manufacturing.

3.1. Experiment settings

In the experiment, subjects wearing a HoloLens 2 perform a specific task on a poster under various conditions manipulated by different factors, including task type, input method, and distance (see Fig. 1). The target selection is confirmed by pressing the space key on the keyboard. Table 1 shows the settings of these factors. Each participant will complete 12 different tasks three times each, resulting in a total of 36 trials. The trial sequence is randomized to avoid potential learning effects. The

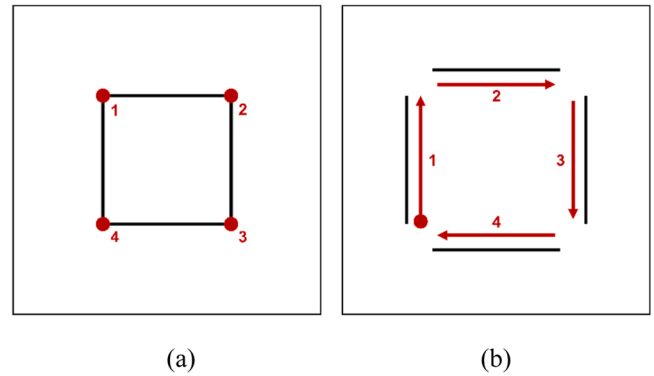


Fig. 2. The targets and sequences for (a) pointing (b) tracing.

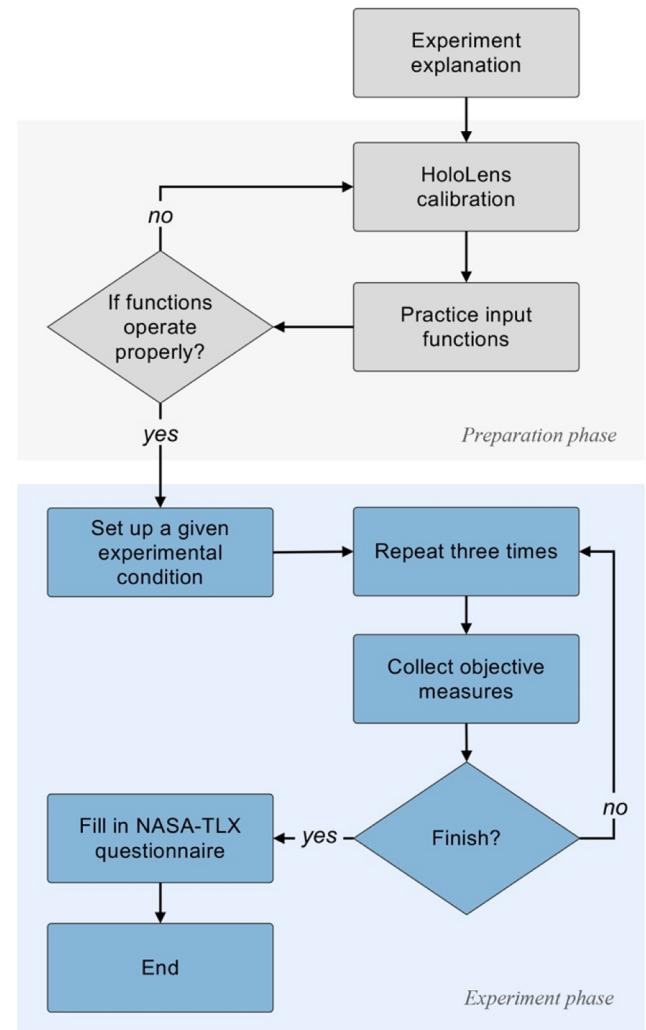


Fig. 3. The experimental procedure.

working principles of the three input methods are as follows.

- **Hand Ray:** Microsoft's Mixed Reality Toolkit (MRTK) v2.0 allows HoloLens users to interact with out-of-reach 2D and 3D content through hand gestures. The hand ray method extends a ray from the user's palm center as an extension of the hand. A cursor at the end of the ray indicates the location where the ray intersects with an object in real environment.

- **Head Gaze:** head gaze represents the direction in which the user's head is pointing. As an input method, a ray is projected forward from HoloLens to determine what it intersects in the real world. The device includes multiple optical and inertial sensors for accurately tracking the position and orientation of the built-in cameras, which remain fixed relative to the user's head.
- **Eye Gaze:** HoloLens 2 provides developers with the ability to use information about what the user is looking at enabled by built-in eye tracking. This introduces a new dimension of human understanding while interacting with an AR scene. Using MRTK, eye gaze is employed as an input method by generating a ray from the gaze origin and along the direction at approximately 30 FPS.

All three input methods determine the target object by intersecting a ray with real-world depth data, which is estimated using HoloLens' depth sensing feature. Alternatively, the ray can intersect with virtual objects within an AR scene. The pointing and tracing targets are located on the poster at two different distances from the subject in the experiment. As shown in Fig. 1, the distance d_1 is varied at two levels: 80 and 130 cm. The distance between the subject and the keyboard remains constant at 30 cm. In this condition, d_2 becomes 50 and 100 cm, respectively. A virtual plane is placed at a specified distance for the ray intersection. The poster is a 60 cm x 60 cm square containing four QR codes near each corner and another one right at the center. As shown in Fig. 2, four dots on the poster form a 25 cm x 25 cm square in the pointing task. In the tracing task, the target consists of four disconnected straight edges, each 25 cm long, forming a 30 cm x 30 cm square. The numbers in the figure specify the clockwise target sequence for subjects to follow.

3.2. Experimental procedure

Fig. 3 shows the procedure for participants to follow in the experiment. The first phase involves preparatory work, including explanation of the experiment objectives, signing the consent document, and calibration of HoloLens for each participant. The calibration step ensures the proper functioning of the input methods by adjusting related parameters based on the position and orientation of individual's headset. A practice session is arranged to familiarize participants with the experiment tasks using HoloLens and the developed interfaces. The tasks in the practice are different from the actual ones to reduce potential learning effects. The actual practice duration ranged from 10 to 15 min varying among participants. In the experimental phase, each subject will complete pointing and line tracing under 12 different conditions, with each condition repeated three times. Objective measures will be collected to evaluate task performance during each trial. After completing the three trials of each condition, participants fill out questionnaires to provide subjective measures. They also have a 2-minute pause before starting the next trial.

3.3. Evaluation criteria

The experiment quantitatively evaluates the task performance in each condition using three criteria of manufacturing implications. First, Acc_P measures the deviation of the target position from the ground truth. In the pointing task, the deviation from the four corners of the square is not differentiated (see Fig. 2(a)). The target position for a corner is estimated as the averaged position of the three trials. The actual position of each vertex on the poster can be directly measured prior to the experiment. The accuracy is estimated as the averaged deviation among each input position with respect to the target. The performance criterion $Prec_P$ measures the precision of a pointing input method. It is determined by averaging the deviation across three trials for each corner.

Likewise, the deviation from the four sides of the square is not differentiated in the tracing task (see Fig. 2(b)). The actual position of each side can be estimated through a calibration procedure between the

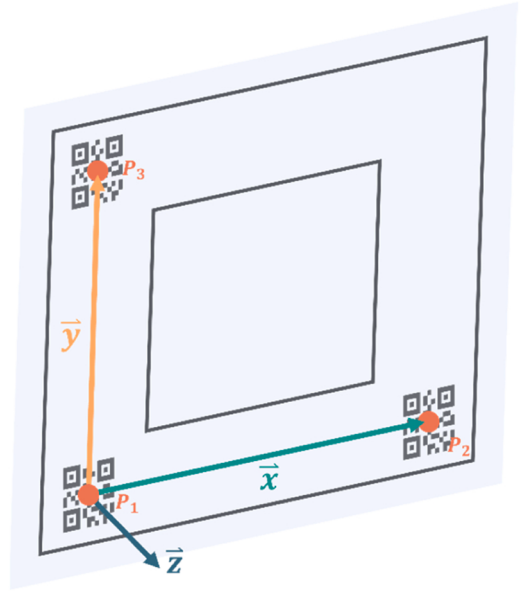


Fig. 4. The calibration procedure based on three QR codes.

HoloLens space and the real environment before conducting the experiment. Acc_T denotes the accuracy of tracing an edge using an input method. First, the average distance of all the input points to an edge is calculated; Acc_T is then estimated as the average deviation among the input points with respect to all four edges. The number of points along one edge may differ from the number along another. Estimating the precision of line tracing starts with constructing a line from the points generated along each edge using the least squares method. $Prec_T$ is calculated by averaging the distances of all input points from the line constructed using the corresponding edges.

In the pointing task, the total time required to specify the four corners is calculated. However, the time spent on moving between edges is not counted in the tracing task. Participants are required to indicate the start and end points with respect to an edge by using the space bar on the keyboard. The task completion time is equal to the average of the three trials in each condition.

3.4. Experimental calibration

The experiment settings involve the coordinate system of the HoloLens and the real world. A calibration procedure based on QR codes is conducted to establish the correlation between both systems. As shown in Fig. 4, a QR code is attached to each of three pre-selected position P_1, P_2, P_3 on the poster. The real-world coordinates of these QR codes are estimated using Vuforia deployed on the HoloLens. An orthogonal coordinate system $(\hat{x}, \hat{y}, \hat{z})$ is determined as:

$$\hat{x} = \frac{P_2 - P_1}{\|P_2 - P_1\|} \quad (1)$$

$$\hat{y} = \frac{P_3 - P_1}{\|P_3 - P_1\|} \quad (2)$$

$$\hat{z} = \hat{x} \times \hat{y} \quad (3)$$

$$M = \begin{bmatrix} \hat{x}_x & \hat{y}_x & \hat{z}_x \\ \hat{x}_y & \hat{y}_y & \hat{z}_y \\ \hat{x}_z & \hat{y}_z & \hat{z}_z \end{bmatrix} \quad (4)$$

$$M_{R-H} = \begin{bmatrix} M & P_1 \\ 0 & 1 \end{bmatrix} \quad (10)$$

All geometric elements used in the experiment are specified within

Table 2
ANOVA results of pointing accuracy.

Source	SS	df	MS	F	p	η_p^2
Distance (D)	90.41	1	90.41	49.49	< .001 *	.303
Input Method (I)	44.84	2	22.42	12.27	< .001 *	.177
D*I	1.87	2	.93	.51	.600	.009
Error	208.25	114	1.82			
Total	3505.36	120				

$(\hat{x}, \hat{y}, \hat{z})$ with its origin at P_3 . Multiplying the 3D location estimated by the HoloLens with the inverse of $M_{R \rightarrow H}$ yields a point in the coordinate system.

4. Experimental results and analysis

Twenty college students with equal gender between the ages of 20 and 25 were recruited to participate in the experiment. Twelve out of the twenty subjects had prior experience using HoloLens 2. The experimental procedure includes a practice session to familiarize all participants with the test tasks using HoloLens and the developed interfaces. Therefore, it is assumed that the difference in AR experience would not influence the experimental results. The pointing and tracing tasks were conducted in a separate session, following the experimental procedure shown in Fig. 3. Each participant had to complete six different conditions in a random sequence and each condition repeated three times. These conditions varied based on three input methods and two target distances in each session. The performance of the experimental tasks

was evaluated using both objective and subjective measures collected during the experiment. The following analyses were conducted on the averaged performance of both the four corners and the four edges.

4.1. Pointing task

4.1.1. Accuracy

There were 120 data points obtained from 20 participants across six different conditions in the experiment. The Kolmogorov-Smirnov test result indicated a normal distribution among these points. The data also passed the Levene test for homogeneity of variance. Table 2 presents the ANOVA results for pointing accuracy. Fig. 5(a) shows that the accuracy at the near distance is significantly higher than that at the far distance. The Tukey HSD post hoc test shows that the eye gaze accuracy is significantly lower than both the hand ray and head gaze methods. The latter two do not exhibit a significant difference (Fig. 5(b)), although the hand ray input has the highest accuracy at 4.47 mm.

4.1.2. Precision

The Kolmogorov-Smirnov test failed to show a normal distribution among the data points. Thus, the Kruskal-Wallis method is applied for non-parametric analysis of the data. According to the analysis result, both the input method and distance have p -values smaller than 0.05, indicating a significant difference in both factors. Fig. 6(a) shows that the precision at the near distance is significantly higher than that at the far distance. The Tukey HSD post hoc test shows that the hand ray precision (1.95 mm) is significantly higher than both the head and eye

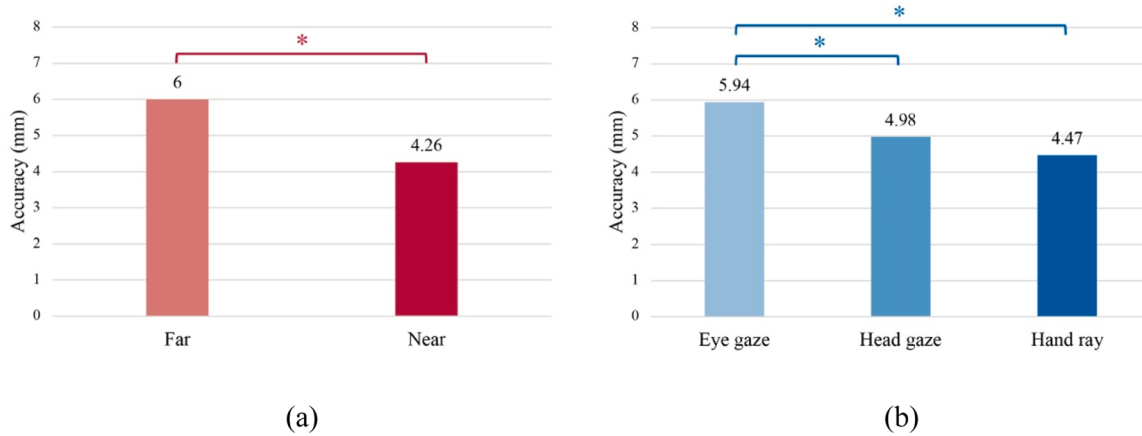


Fig. 5. Pointing accuracy with respect to (a) distance and (b) input method.

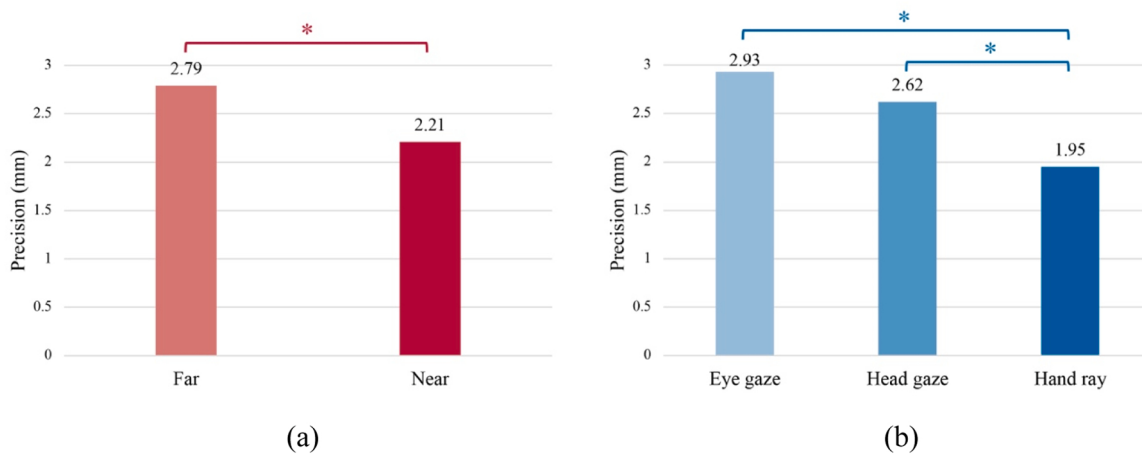


Fig. 6. Pointing precision with respect to (a) distance and (b) input method.

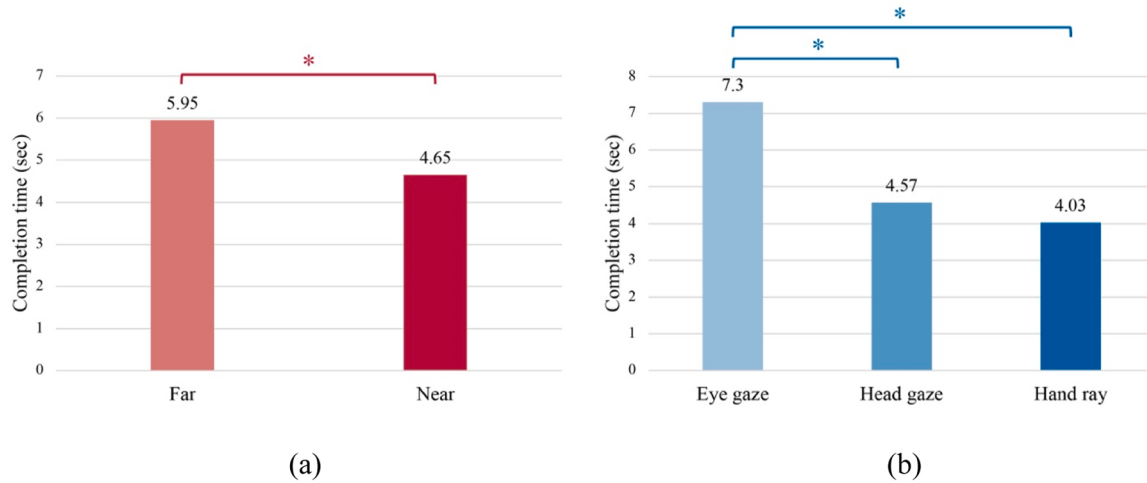


Fig. 7. Completion time of pointing with respect to (a) distance and (b) input method.

Table 3

ANOVA results of tracing accuracy.

Source	SS	df	MS	F	p	η_p^2
Distance (D)	148.508	1	148.508	40.587	< .001 *	.103
Input Method (I)	1123.917	2	561.958	153.582	< .001 *	.465
D*I	1.913	2	.956	.261	.770	.001
Error	1295.293	354	3.659			
Total	11964.996	360				

gaze methods. The latter two do not exhibit a significant difference as shown in Fig. 6(b).

4.1.3. Completion time

The Kolmogorov-Smirnov test failed to show a normal distribution among the data points. Thus, the Kruskal-Wallis method is applied for non-parametric analysis of the data. According to the analysis result, both the input method and distance have p -values smaller than 0.05, indicating a significant difference in both factors. The Dunn's post hoc test shows that the completion time at the near distance is significantly shorter than that at the far distance (see Fig. 7(a)). The pointing task is more efficient at the near distance. The completion time of the eye gaze input is significantly longer than that of both the hand ray and head gaze methods, while the latter two do not exhibit a significant difference (Fig. 7(b)).

4.2. Line tracing task

4.2.1. Accuracy

The sample size of tracing accuracy is sufficiently large to characterize the statistical performance of the population. Therefore, it is not necessary to conduct normality and homogeneity of variance tests on the data. Table 3 presents the ANOVA results for the experimental conditions varied by the distance and input method. The Tukey HSD post hoc test shows that the accuracy at the near distance is significantly higher than that at the far distance (see Fig. 8(a)). The eye gaze accuracy is significantly lower than both the hand ray and head gaze methods (see Fig. 8(b)), while the latter two do not exhibit a significant difference.

4.2.2. Precision

For the same reason as accuracy, it is not necessary to conduct

Table 4

ANOVA results of tracing precision.

Source	SS	df	MS	F	p	η_p^2
Distance (D)	28.310	1	28.310	43.416	< .001 *	.109
Input Method (I)	5.364	2	2.682	4.113	.017 *	.023
D*I	.773	2	.386	.593	.553	.003
Error	230.833	354	.652			
Total	2464.668	360				

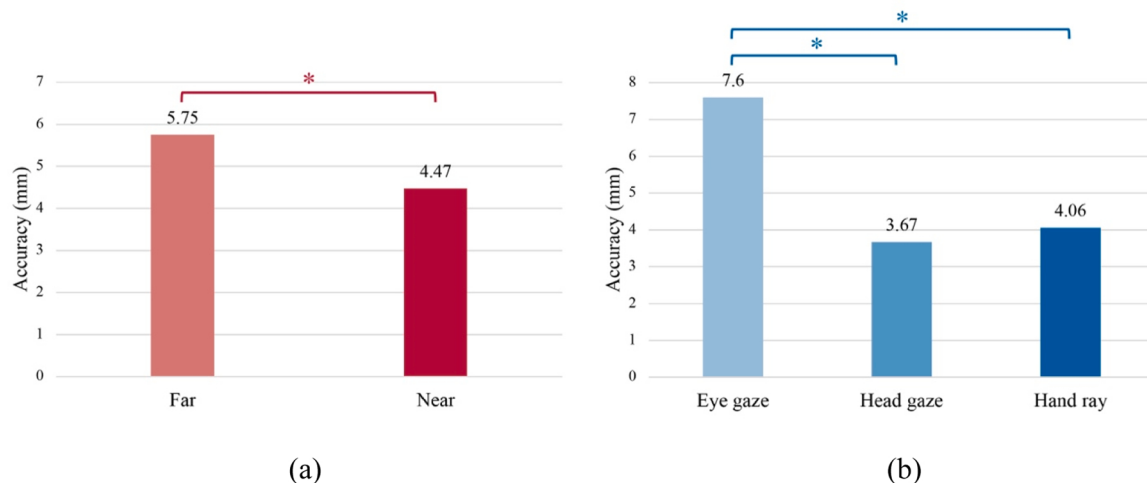


Fig. 8. Line tracing accuracy with respect to (a) distance and (b) input method.

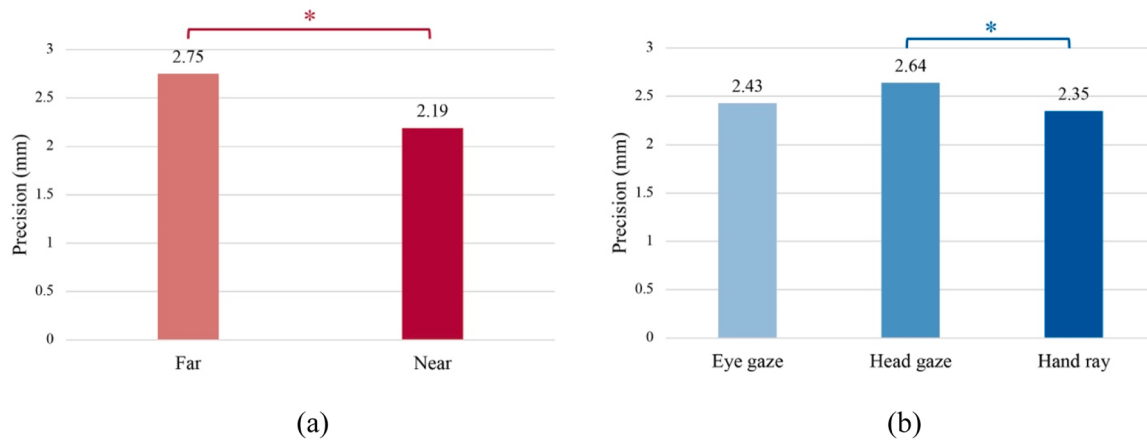


Fig. 9. Line tracing precision with respect to (a) distance and (b) input method.

Table 5
ANOVA results of tracing time.

Source	SS	df	MS	F	p	η_p^2
Distance (D)	148.508	1	148.508	40.587	< .001 *	.103
Input Method (I)	1123.917	2	561.958	153.582	< .001 *	.465
D*I	1.913	2	.956	.261	< .001 *	.001
Error	1295.293	354	3.659			
Total	11,964.996	360				

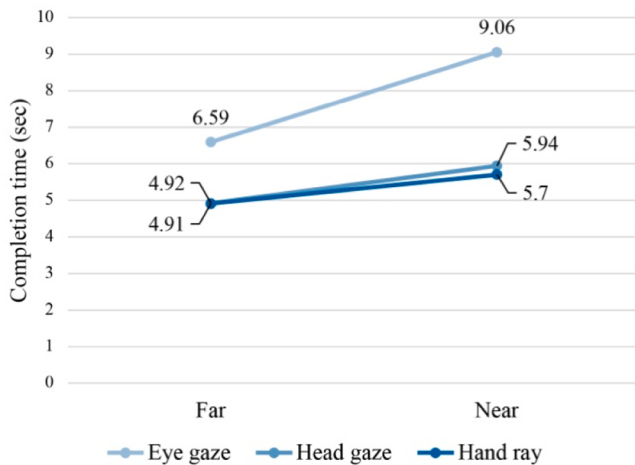


Fig. 10. Analyzing interaction effects on the completion time of line tracing.

normality and homogeneity of variance tests for tracing precision. Table 4 presents the ANOVA results for the experimental conditions varied by the distance and input method. The Tukey HSD post hoc test shows that the precision at the near distance is significantly higher than that at the far distance (see Fig. 9(a)). The hand ray precision is significantly higher than that of the head gaze method (see Fig. 9(b)). Regardless of the experimental conditions, the precision value typically falls within the range of 2 to 3 mm.

4.2.3. Completion time

For the same reason as accuracy, it is not necessary to conduct normality and homogeneity of variance tests for completion time. Table 5 presents the ANOVA results for the experimental conditions varied by the distance and input method. According to the table, there is a significant interaction between the two factors, distance and input method, which differs from all previous analyses. Fig. 10 shows how the

completion time is influenced by the interaction. Line tracing at the near distance requires less time than tracing at the far distance. Eye gaze input takes more time than the other two input methods. Note that the completion time mentioned here refers to the total duration required to complete the tracing of all four edges.

4.3. Subjective evaluation

A standard SUS questionnaire [47] was employed to evaluate the usability of the three input methods implemented in AR. This work skipped the pairwise comparisons to determine the weights associated with the six dimensions. Therefore, the assessment test conducted with this simplification is referred to as “raw TLX”. Participants rated each question in the questionnaire on a five-point Likert scale after completing three trials of an experimental condition. The Kruskal-Wallis method was first applied for non-parametric analysis of the 120 data points generated by 20 subjects under 6 conditions. The result indicates a significant difference among the input methods. The Dunn’s post hoc test shows that the score of the hand ray method is significantly higher than that of both head and eye gaze, while the latter two scores do not exhibit a significant difference (see Fig. 11(a)). Only the hand ray is considered above average with a score of 80.19, while the other methods are below average.

Additionally, the NASA-TLX questionnaire [48] was used for subjective assessment of the 120 data points. The test result of the Kruskal-Wallis method indicates a significant difference among the input methods. Therefore, the Dunn’s test was applied for post hoc analysis. The result shows that the workload induced by the hand ray method is significantly lower than that induced by both head and eye gaze, while the latter two do not exhibit a significant difference (see Fig. 11(b)). Fig. 12 compares the three methods based on six subjective subscales. The hand ray input outperforms the other methods in each subscale.

4.4. Discussion

This section summarizes key findings obtained from the statistical analysis of the experimental data presented in the previous sections. Possible explanations are also provided to discuss the implications of these findings. Note that the experimental condition in which the subjects were seated differs from the operational environment of a welding robotic arm, where the human operator is typically standing on the shop floor. It is assumed that the difference between the sitting and standing postures will not influence the comparison result, specifically, it will not change the performance rank among the three input methods.

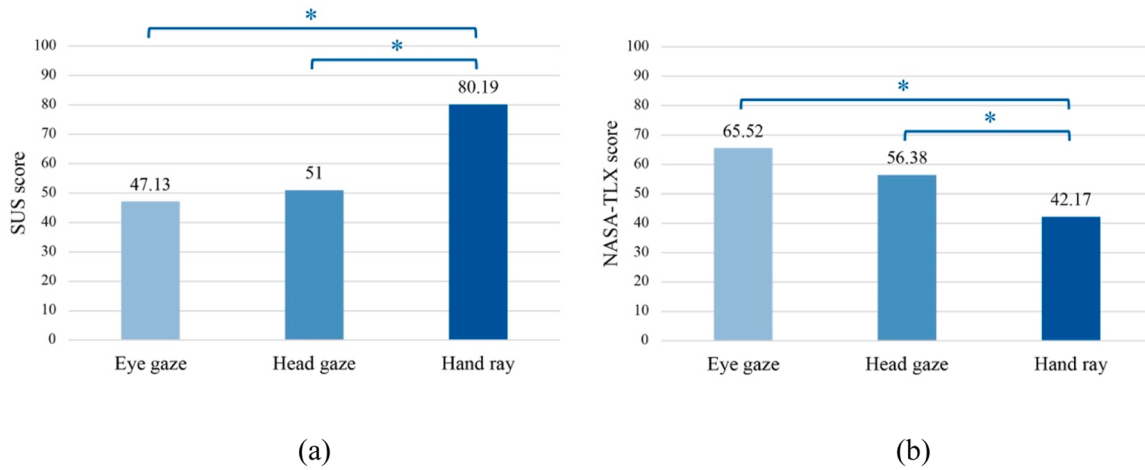


Fig. 11. Subjective assessment results using (a) SUS and (b) NASA-TLX.

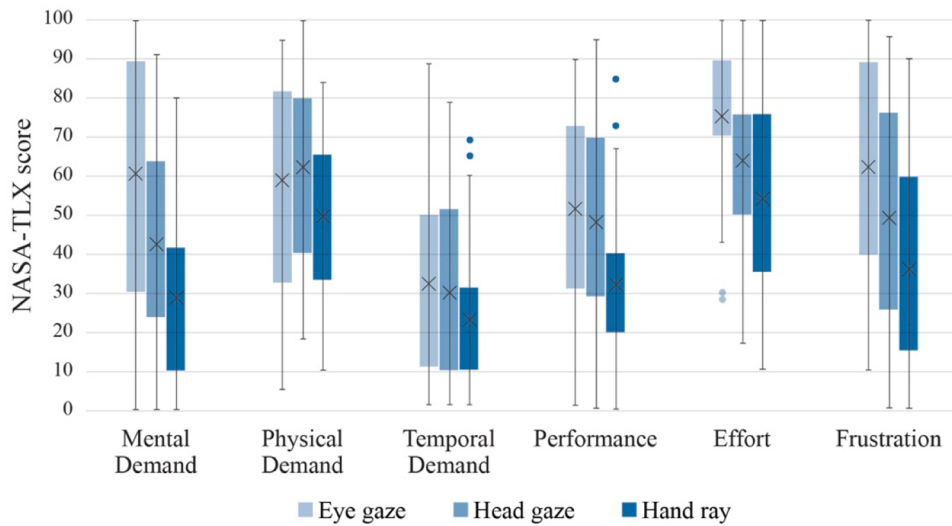


Fig. 12. Analysis of subscales in NASA-TLX.

- Overall, the hand ray and head gaze methods outperform the eye gaze method in terms of accuracy, precision, and completion time for both pointing and tracing tasks at both distances. Both SUS and NASA-TLX scores indicate that participants prefer using the hand ray over the head gaze method due to its lower workload and higher usability. Hence, the hand ray input method is chosen for the prototyping AR tool developed for robotic welding programming.
- The highest accuracy achieved in pointing and line tracing is 4.47 mm and 3.67 mm, while the finest precision is 1.95 mm and 2.35 mm, respectively. These performances would not meet the requirements of fine movement control. The positional tolerance in AR-assisted operations that involve specifying points in the real world should be reasonably generous. AR applications implemented only with commercial HMDs nowadays may not effectively support manufacturing tasks requiring submillimeter accuracy, such as precision machining and assembly. However, they can be employed to assist other industrial activities that require less stringent positional accuracy like spray painting [49], order picking [50], and assembly guidance [51].
- Based on the post-experiment interview, participants occasionally experienced eye fatigue due to frequent adjustments of their eye gaze during the experiment. Repetitive head movement for fine control also caused their neck pain or stiffness. The lower SUS and NASA-

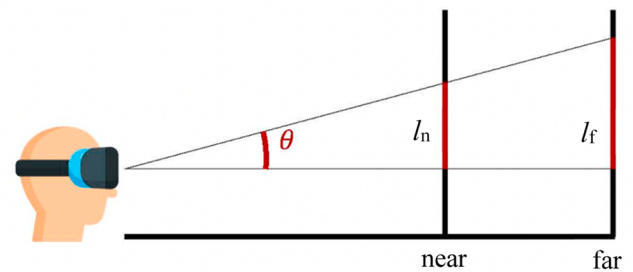


Fig. 13. Rotation results in a larger point movement over a greater distance in the real world.

TLX scores of the head and eye gaze methods support these observations. Only the usability of the hand ray was considered as acceptable.

- The target distance influences the completion time of pointing and tracing tasks differently. Pointing to a distant location takes more time than pointing to a nearby one. In contrast, line tracing at a greater distance is faster than that at a shorter distance. All three input methods involve rotating specific body parts for precise control

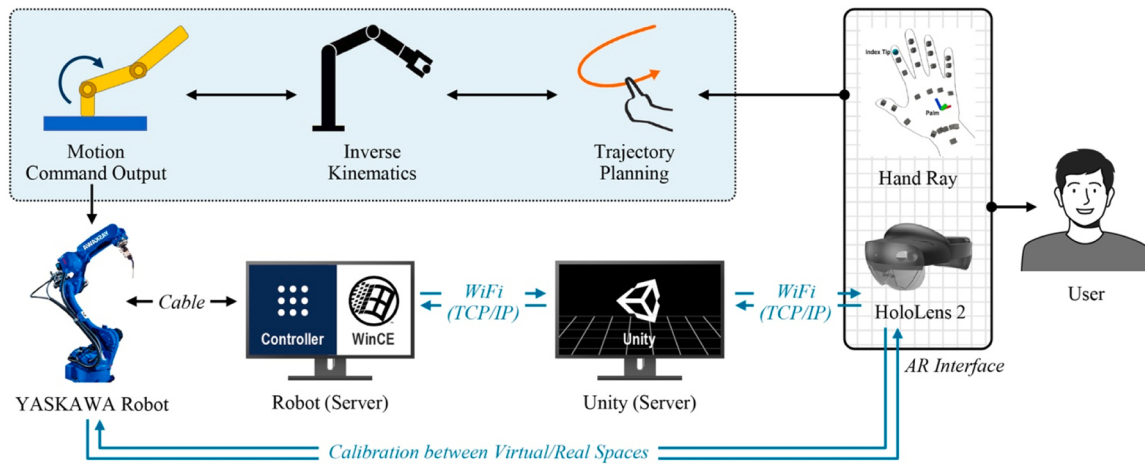


Fig. 14. The system framework of the prototyping AR tool.

of the input point. As shown in Fig. 13, the same rotation can result in a larger point movement over a greater distance in the real world. In this case, participants tended to mark fewer points along the traced line, producing a shorter completion time. However, the larger point movement may make precise input adjustments to the target in the pointing task more difficult and time-consuming.

- Note that the eye and head gaze methods provided by modern AR headsets were originally developed for objection selection, such as aiming or triggering a button press [52], rather than high-precision tasks. The experimental results of this study confirm the previous conclusion that tracking gaze alignment over a target is not precise or easy [53]. The current gaze methods require a camera-based calibration process to correlate eye images with gaze directions. The deviation between the actual and estimated gaze directions can be substantial due to poor calibrations. Moreover, uncontrollable physiological factors in the human body, such as breathing, hand movements, eye micro-saccade [54], and head jitter [55], introduce a level of noise within gaze signals and the focal point. Algorithms specifically designed to enhance pointing and tracing tasks in robot PbD are required to reduce those errors through gaze dispersion or velocity thresholds.

5. Prototyping system of robotic mold welding

5.1. Robotic welding of tire mold

Rubber tires are mainly produced by injection molding under high pressure within a set of segmented molds arranged in a circular configuration. Referred to as tire molds, these molds are typically made of aluminum or steel and used for thousands of times in tire production. A tire mold could suffer various damages during use, such as deformation, wear, and surface peeling, which require repair to prolong its lifespan. A common procedure for rectifying these damages starts with adding extra material to the damaged area by robotic welding. To fully automate the robot programming in the mold welding is difficult for two main reasons. First, robot motion planning depends on precise recognition of damage properties, including type, geometry, and location, which may be problematic in real-world deployment. Additionally, the complete CAD model of a segmented tire mold is often not available on the shop floor [56]. Programming by demonstration would be a more practical approach for robot motion planning under such circumstances.

5.2. System framework

A prototyping AR tool was developed for this purpose to show the practicality of the hand ray input method in real manufacturing. Fig. 14 shows the system framework of the AR tool. A calibration procedure

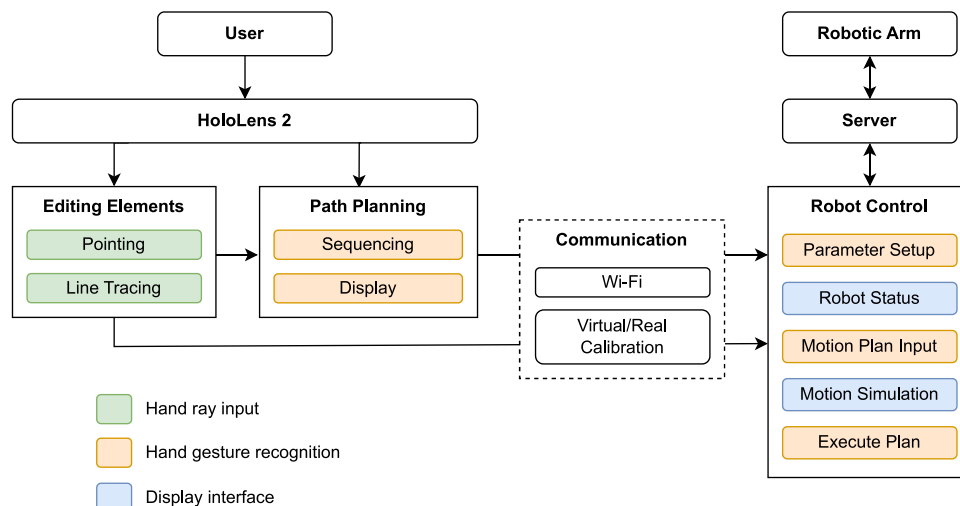


Fig. 15. The user interfaces provided by the prototyping AR tool.

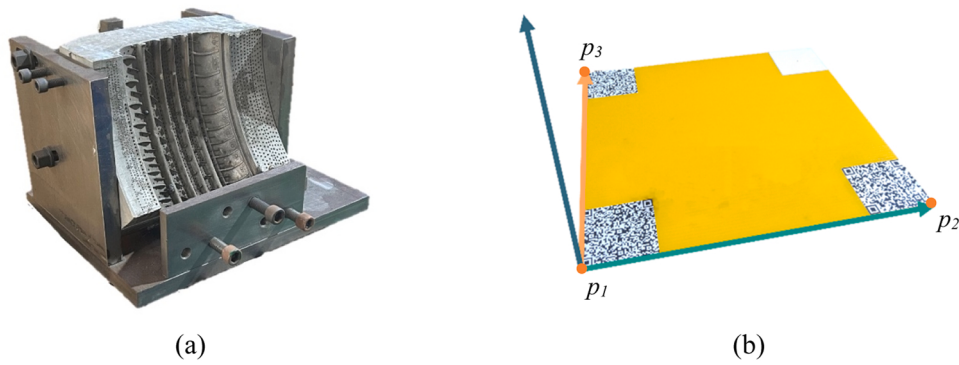


Fig. 16. The tire mold welding setup: (a) a fixture and (b) a calibration plate.

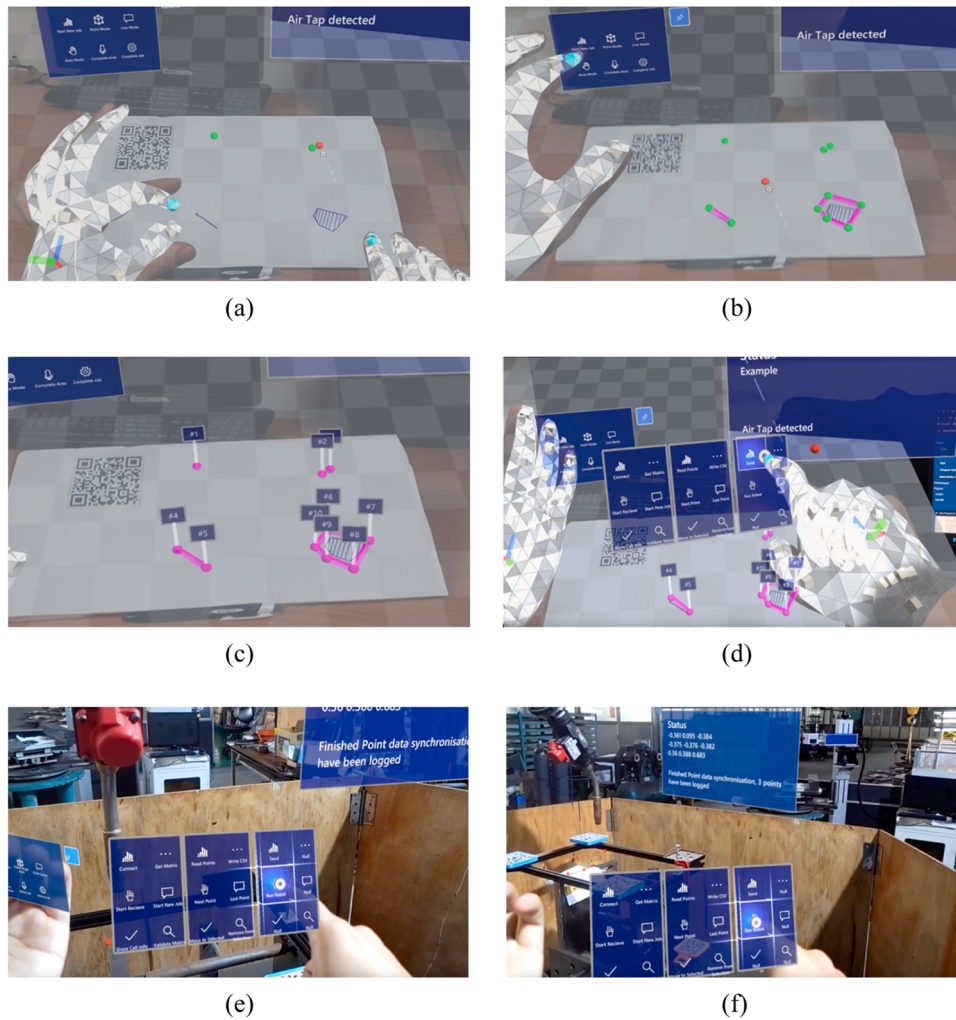


Fig. 17. Various interactions with the prototyping AR tool: (a) specifying points on a plane, (b) constructing geometric elements, (c) labeling constructed elements, (d), storing the constructed elements (e) conducting dry run through hand gestures, and (f) the dry run process.

based on QR codes was first conducted to align the HoloLens and robot coordinate systems. The hand ray input was implemented using MRTK in HoloLens 2. Robot motion planning mainly consists of three functional modules. The Trajectory Planning Module offers geometric processing functions and user interfaces to complete pointing and line tracing. The Inverse Kinematics Module transforms the planning result into the motion commands of each joint in a YASKAWA 6-DOF robotic arm. This module also calculates all possible robot postures for achieving the motion and highlights those without singularity points. It

was deployed on the robot server, which communicates with the Unity server via Wi-Fi using TCP/IP. The robot controller running on the same server continuously drives the robot's motions via a cable. This is accomplished in the third module by converting inverse kinematics results into motion commands specific to the robotic arm. This module also determines and manifests the process parameters related to the welding operation in the commands.

As shown in Fig. 15, the prototyping AR tool consists of three main interactive functions: Editing Elements, Path Planning, and Robot

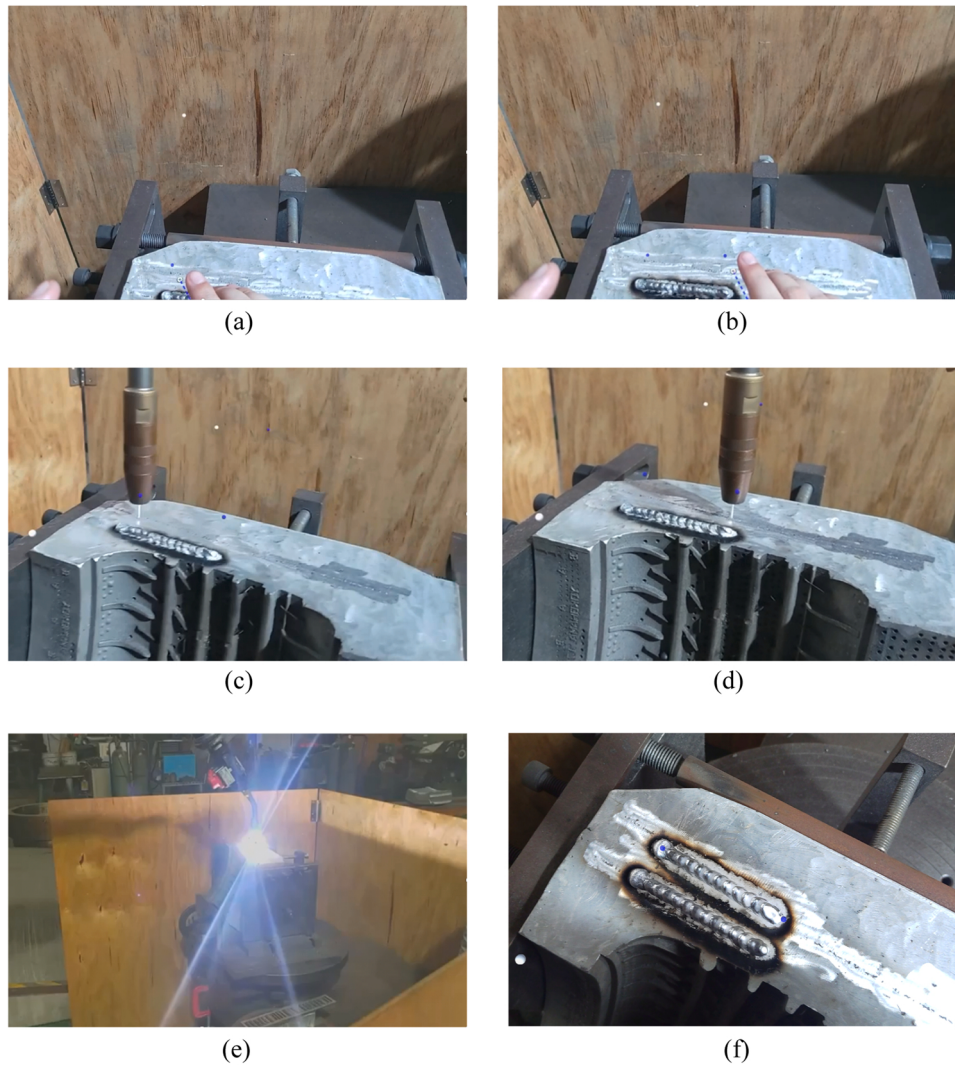


Fig. 18. The robotic welding process based on pointing: (a) specifying the first point, (b) specifying the second point, (c) dry run at the first point, (d) dry run at the second point, (e) actual welding, and (f) the welding result.

Control. A human planner can access these functions through the user interfaces deployed on a HoloLens HMD. The interfaces can be interacted with using three different methods: hand ray, hand gesture recognition, and visual display only. These methods are indicated with different colors in the figure. The instant communication between the HoloLens and robotic arm is achieved through a wireless network. The coordinate systems between the virtual and real worlds have been precisely aligned through a calibration procedure based on QR codes.

5.3. Test result

The prototyping AR tool was applied to a real scenario of robotic welding for tire mold repair. A fixture was designed to hold a segmented tire mold in place, as shown in Fig. 16. A plate containing QR codes at three corners was used in a calibration procedure similar to the principle explained in Section 3.4. The following images were acquired using HoloLens from the first-person viewpoint. Fig. 17 illustrates various



Fig. 19. Deviations of line tracing in the robotic welding process: (a) hand ray input and (b) the welding result on the tire mold.

interaction results during AR-assisted robot programming using the hand ray input and hand gestures. The user can specify basic geometric elements on a plane through pointing and line tracing, label the specified elements, and store them on the robot server. The HoloLens also enables direct dry run execution through hand gestures in AR. Suppose a layer of material was added on the top of the tire mold during the welding process. As shown in Fig. 18(a) and (b), the user first selected two points to define the welding path on the top surface using the hand ray input method. The point locations in the real-world were estimated based on the transformation matrix established in the calibration procedure. The robot tip followed the specified point locations while maintaining a safety distance during a dry run (Fig. 18(c) and (d)). Once the welding path was verified, the system guided the robot to perform the actual welding as shown in Fig. 18(e). Fig. 18(f) shows the final welding result generated along this path on the tire mold. Users could also specify a welding path using the line tracing input method following a similar procedure. Fig. 19 shows the test results in this case. The actual path was constructed by linear regression of the discrete points obtained during the tracing process. It is evident that some points have a significant deviation from the constructed path.

6. Conclusion

Industrial robots are an essential component that realizes the idea of Industry 4.0 on the shop floor. Traditional robot programming involves either a model construction process or generation of motion commands using a teaching pendant, both of which are time-consuming and lack flexibility. AR technology provides engineers with effective user interfaces featuring various modalities to assist in robot planning, particularly through the approach of programming by demonstration. This research conducted an experimental study to evaluate three common input methods in modern AR, with a focus on their applicability in the manufacturing context. The study cross-compared hand ray, head gaze, and eye gaze inputs for pointing and line tracing tasks at varying distances based on both subjective and objective measures. In most experimental conditions, the hand ray and head gaze outperform the eye gaze method in terms of accuracy, precision, and completion time. However, both SUS and NASA-TLX scores indicated that participants prefer using the hand ray over the head gaze method. The positional accuracy and precision achieved in pointing and line tracing through the experimental AR interfaces would not meet the requirements of fine movement control in manufacturing. Pointing takes more time at a greater distance, whereas line tracing is faster at a greater distance than at a shorter one. The experimental findings obtained by this study confirm previous research that tracking gaze alignment over a target is imprecise using modern AR headsets. The primary challenge is to overcome and compensate for uncontrollable physiological factors in the human body that can cause deviations in gaze signals and the focal point. In addition, a prototyping AR tool was developed to demonstrate the feasibility of the hand ray method as a non-contact input interface for robot PbD in mold welding. A system framework describes functional features provided by the AR tool, including trajectory planning, inverse kinematics, and generation of motion commands implemented using Unity and ROS servers. An industrial robotic arm completed welding operations on a metal tire mold by following the motions specified through pointing and line tracing on the mold surface. The test results have verified the feasibility of robot programming on real objects using AR interfaces. This work demonstrates an exemplary application of human-robot collaboration in real manufacturing.

Non-contact pointing or tracing with HoloLens results in significant positional deviations that do not meet the precision requirements of most manufacturing processes. Future work can explore determining the precise position of an input point using an external sensor, rather than direct use of an AR headset. Incorporating sensory cues like haptic force may also improve precision and efficiency of the current input methods controlled by body motion.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was financially support by Minister of Science and Technology Taiwan under the grant number 2221-E-007-098-MY3.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.jmsy.2024.03.016.

References

- [1] Chu CH, Wang L, Liu S, Zhang Y, Menozzi M. Augmented reality in smart manufacturing: enabling collaboration between humans and artificial intelligence. *J Manuf Syst* 2021;61:658–9.
- [2] Baroroh DK, Chu CH, Wang L. Systematic literature review on augmented reality in smart manufacturing: collaboration between human and computational intelligence. *J Manuf Syst* 2021;61:696–711.
- [3] Li S, Wang R, Zheng P, Wang L. Towards proactive human–robot collaboration: a foreseeable cognitive manufacturing paradigm. *J Manuf Syst* 2021;60:547–52.
- [4] Green SA, Billingham M, Chen X, Chase JG. Human-robot collaboration: a literature review and augmented reality approach in design. *Int J Adv Robot Syst* 2008;5(1):1.
- [5] Hietanen A, Pieters R, Lanz M, Latokartano J, Kämäräinen JK. AR-based interaction for human-robot collaborative manufacturing. *Robot Comput-Integr Manuf* 2020; 63:101891.
- [6] Chu CH, Liu YL. Augmented reality user interface design and experimental evaluation for human-robot collaborative assembly. *J Manuf Syst* 2023;68:313–24.
- [7] Mourtzis D, Angelopoulos J, Panopoulos N. Operator 5.0: a survey on enabling technologies and a framework for digital manufacturing based on extended reality. *J Mach Eng* 2022;22.
- [8] Behere, S.2010. A generic framework for robot motion planning and control.
- [9] Blankemeyer S, Wiemann R, Posniak L, Pregizer C, Raatz A. Intuitive robot programming using augmented reality. *Procedia CIRP* 2018;76:155–60.
- [10] Chu CH, Liu YW, Li PC, Huang LC, Luh YP. Programming by demonstration in augmented reality for the motion planning of a three-axis CNC dispenser. *Int J Precis Eng Manuf-Green Technol* 2020;7:987–95.
- [11] Ong SK, Yew AWW, Thanigaivel NK, Nee AY. Augmented reality-assisted robot programming system for industrial applications. *Robot Comput-Integr Manuf* 2020;61:101820.
- [12] Elsdon J, Demiris Y. Augmented reality for feedback in a shared control spraying task. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*. IEEE; 2018, May. p. 1939–46.
- [13] Dhanaraj N, Yoon YJ, Malhan R, Bhatt PM, Thakar S, Gupta SK. A mobile manipulator system for accurate and efficient spraying on large surfaces. *Procedia Comput Sci* 2022;200:1528–39.
- [14] Zaeh MF, Vogl W. Interactive laser-projection for programming industrial robots. In: *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE; 2006, October. p. 125–8.
- [15] Fang HC, Ong SK, Nee AY. Robot path and end-effector orientation planning using augmented reality. *Procedia CIRP* 2012;3:191–6.
- [16] Veiga G, Malaca P, Cancela R. Interactive industrial robot programming for the ceramic industry. *Int J Adv Robot Syst* 2013;10(10):354.
- [17] Fang HC, Ong SK, Nee AY. Novel AR-based interface for human-robot interaction and visualization. *Adv Manuf* 2014;2:275–88.
- [18] Araque D, Díaz R, Pérez-Gutiérrez B, Uribe AJ. Augmented reality motion-based robotics off-line programming. In: *Proceedings of the IEEE Virtual Reality Conference*. IEEE; 2011, March. p. 191–2.
- [19] Pai YS, Yap HJ, Singh R. Augmented reality-based programming, planning and simulation of a robotic work cell. *Proc Inst Mech Eng, Part B J Eng Manuf* 2015;229(6):1029–45.
- [20] Norberto Pires J, Veiga G, Araújo R. Programming by demonstration in the coworker scenario for SMEs. *Ind Robot Int J* 2009;36(1):73–83.
- [21] Frank JA, Moorhead M, Kapila V. Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks. In: *Proceedings of the twenty fifth IEEE int symp robot hum interact commun (RO-MAN)* 2016:302–7.
- [22] Chacko SM, Kapila V. An augmented reality interface for human-robot interaction in unconstrained environments. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE; 2019. p. 3222–8.
- [23] De Pace F, Sanna A, Manuri F, Oriti D, Panucci S, Belcamino V. Assessing the effectiveness of augmented reality handheld interfaces for robot path programming. In: *Proceedings of the international conference on intelligent technologies for interactive entertainment*. Cham: Springer International Publishing; 2021. p. 336–52.

- [24] Hügler J, Lambrecht J, Krüger J. An integrated approach for industrial robot control and programming combining haptic and non-haptic gestures. In: Proceedings of the twenty sixth IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE; 2017. p. 851–7.
- [25] Kapinus M, Beran V, Materna Z, Bambušek D. Spatially situated end-user robot programming in augmented reality. In: Proceedings of the twenty eighth IEEE international conference on robot and human interactive communication (RO-MAN). IEEE; 2019. p. 1–8.
- [26] Ostanin M, Klimchik A. Interactive robot programming using mixed reality. IFAC-Pap 2018;51(22):50–5.
- [27] Puljiz, D., Hein, B. 2019. Concepts for end-to-end augmented reality based human-robot interaction systems. Available from: arXiv preprint arXiv:1910.04494.
- [28] Yigitbas E, Jovanovikj I, Engels G. Simplifying robot programming using augmented reality and end-user development. In: Proceedings of the eighteenth IFIP TC 13 international conference, Bari, Italy, August 30–September 3, 2021, Part I, 18. Springer Publishing; 2021. p. 631–51.
- [29] Solanes JE, Munoz A, Gracia L, Martí A, Gírbés-Juan V, Tórner J. Teleoperation of industrial robot manipulators based on augmented reality. Int J Adv Manuf Technol 2020;111:1077–97.
- [30] Lambrecht J, Krüger J. Spatial programming for industrial robots based on gestures and augmented reality. IEEE/RSJ Int Conf Intell Robots Syst 2012;2012:466–72.
- [31] Araiza-Illan D, De San Bernabe A, Hongchao F, Shin LY. Augmented reality for quick and intuitive robotic packing re-programming. pp. 664–664. In: Proceedings of the fourteenth ACM/IEEE international conference on human-robot interaction (HRI). IEEE; 2019.
- [32] Rudorfer M, Guhl J, Hoffmann P, Krüger J. Holo pick'n'Place. In: Proceedings of the IEEE twenty third international conference on emerging technologies and factory automation (ETFA), 1. IEEE; 2018. p. 1219–22.
- [33] Yan Z, Wu Y, Li Y, Shan Y, Li X, Hansen P. Design eye-tracking augmented reality headset to reduce cognitive load in repetitive parcel scanning task. IEEE Trans Hum-Mach Syst 2022;52(4):578–90.
- [34] Chan, W.P., Crouch, M., Hoang, K., Chen, C., Robinson, N., Croft, E. 2022. Design and implementation of a human-robot joint action framework using augmented reality and eye gaze. Available from: arXiv preprint arXiv:2208.11856.
- [35] LaViola Jr, J.J., Kruijff, E., McMahan, R.P., Bowman, D., Poupyrev, I.P. 2017. 3D user interfaces: theory and practice. Addison-Wesley Professional.
- [36] Brewster S, Lumsden J, Bell M, Hall M, Tasker S. Multimodal 'eyes-free' interaction techniques for wearable devices. In: Proceedings of the SIGCHI conference on Human factors in computing systems 2003:473–80.
- [37] Arevalo Arboleda S, Rücker F, Dierks T, Gerken J. Assisting manipulation and grasping in robot teleoperation with augmented reality visual cues. In: Proceedings of the 2021 CHI conference on human factors in computing systems 2021:1–14.
- [38] Chan WP, Quintero CP, Pan MK, Sakr M, Van der Loos HM, Croft E. A multimodal system using augmented reality, gestures, and tactile feedback for robot trajectory programming and execution. In: Virtual reality. River Publishers; 2022. p. 142–58.
- [39] Hoang KC, Chan WP, Lay S, Cosgun A, Croft EA. Arviz: an augmented reality-enabled visualization platform for ros applications. IEEE Robot Autom Mag 2022; 29(1):58–67.
- [40] Sita E, Studley M, Dailami F, Pipe A, Thomassen T. Towards multimodal interactions: robot jogging in mixed reality. In: Proceedings of the twenty third ACM symposium on virtual reality software and technology 2017:1–2.
- [41] Bernardos AM, Gómez D, Casar JR. A comparison of head pose and deictic pointing interaction methods for smart environments. Int J Hum-Comput Interact 2016;32(4):325–51.
- [42] Lin CJ, Ho SH, Chen YJ. An investigation of pointing postures in a 3D stereoscopic environment. Appl Ergon 2015;48:154–63.
- [43] Bates R, Istance HO. Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices. Univers Access Inf Soc 2003;2:280–90.
- [44] Kytö M, Ens B, Piumsomboon T, Lee GA, Billingham M. Pinpointing: precise head- and eye-based target selection for augmented reality. In: Proceedings of the CHI conference on human factors in computing systems 2018:1–14.
- [45] Condino S, Fida B, Carbone M, Cerenelli L, Badiali G, Ferrari V, et al. Wearable augmented reality platform for aiding complex 3D trajectory tracing. Sensors 2020; 20(6):1612.
- [46] Krupke D, Steinicke F, Lubos P, Jonetzko Y, Görner M, Zhang J. Comparison of multimodal heading and pointing gestures for co-located mixed reality human-robot interaction. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE; 2018. p. 1–9.
- [47] Brooke J. Sus: a "quick and dirty" usability. Usability Eval Ind 1996;189(3):189–94.
- [48] Hart SG, Staveland LE. Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In: Advances in psychology, 52. North-Holland; 1988. p. 139–83.
- [49] Elsdon J, Demiris Y. Augmented reality for feedback in a shared control spraying task. In: Proceedings of the IEEE international conference on robotics and automation (ICRA). IEEE; 2018. p. 1939–46.
- [50] Husár J, Knapčíková L. Possibilities of using augmented reality in warehouse management: a study. Acta Logist 2021;8(2):133–9.
- [51] Chu CH, Ko CH. An experimental study on augmented reality assisted manual assembly with occluded components. J Manuf Syst 2021;61:685–95.
- [52] Blattgerste J, Renner P, Pfeiffer T. Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views. Proc Workshop Commun Gaze Interact 2018:1–9.
- [53] Sidenmark L, Parent M, Wu CH, Chan J, Glueck M, Wigdor D, et al. Weighted pointer: error-aware gaze-based interaction through fallback modalities. IEEE Trans Vis Comput Graph 2022;28(11):3585–95.
- [54] Carpenter, R.H. (1988). Movements of the Eyes. 2nd Rev. Pion Limited.
- [55] Sidenmark L, Gellersen H. Eye&head: synergetic eye and head movement for gaze pointing and selection. In: Proceedings of the thirty second annual ACM symposium on user interface software and technology 2019:1161–74.
- [56] Chu CH, Song MC, Luo VC. Computer aided parametric design for 3D tire mold production. Comput Ind 2006;57(1):11–25.