

Full length article

Augmented reality-based robot teleoperation system using RGB-D imaging and attitude teaching device

Yong Pan^a, Chengjun Chen^{a,*}, Dongnian Li^a, Zhengxu Zhao^a, Jun Hong^b^a School of Mechanical and Automotive Engineering, Qingdao University of Technology, Qingdao, Shandong 266000, China^b School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, Shanxi 710049, China

ARTICLE INFO

Keywords:

Robot teleoperation
Online teaching programming
Augmented reality
RGB-D imaging
AR registration

ABSTRACT

Augmented reality (AR)-based programming using the demonstration method has been widely studied. However, studies on AR-based programming for remote robots are lacking because of the limitation of human–computer interaction. This paper proposes an AR-based robot teleoperation system and method using RGB-D imaging and an attitude teaching device. By sending the color and depth images of the remote robot environment to the local side, the operators can complete the teleoperation of the robot at the local side. First, the operators select key positions on the motion path of the robot endpoint from color images via a mouse, and the computer calculates the 3D coordinates of these key points in the robot base coordinate system to complete the position teaching process. In the robot attitude teaching process, the AR technology is used to superimpose the virtual robot model onto the color images of the robot teleoperation environment, so as to make the virtual robot endpoint to move along the teaching path. An operator can use the portable attitude teaching device designed in this study to control the robot movement parameters, such as the attitude and motion speed, during the movement of the virtual robot. After the position and attitude teaching processes, the robot movement trajectory can be generated. To make the base coordinate system of the virtual model consistent with that of the physical robot, we propose an online AR registration method, which does not require manually placing the AR registration marker. The proposed AR-based robot teleoperation system can quickly and easily complete robot teleoperation at the local side.

1. Introduction

Teleoperation involves deploying remote robots to complete dangerous, inaccessible, or delicate tasks [1]. Efficient programming and accurate movement planning of such robots are bottlenecks in manipulating and programming remote robots. Robot programming is a key factor affecting the production efficiency in manufacturing fields [2]. Currently, online lead-through and offline programming methods are widely used in the industry. In the online lead-through programming method, the operator leads a physical robot through its desired motions. The robot can be programmed by recording the states of the encoders mounted on its joints. In online lead-through programming, the physical robot is occupied, which reduces the productivity of the robot. Offline programming requires an extensive modeling process to capture the underlying physical environment in 3D. The modeling process in turn requires a great deal of skill and finesse and is therefore time-consuming and costly.

Programming by demonstration (PbD) is a process used to transfer

new skills to a machine by relying on user demonstrations. PbD is intended to make programming accessible to novice users by providing them with an intuitive and familiar interface [3]. Therefore, human–robot interaction is an important factor affecting the teaching performance of PbD. Currently, demonstration approaches for PbD fall into three categories: kinesthetic teaching, teleoperation, and passive observation [4]. In kinesthetic teaching, a user can program a robot by physically moving it through the desired motions [5], and the robot program can be generated by recording demonstrations directly on the robot using its integrated sensors. In teleoperation demonstration, a user operates robots indirectly using external input (such as a joystick, graphical user interface, and force feedback device). Unlike kinesthetic teaching, teleoperation does not require the user to be in the robot environment, making it suitable for remote robots [6]. Passive observation is also referred to as imitation learning. In this type of demonstration method, a user manually performs a task by himself/herself, and the external sensors (such as a tracking system and machine vision system) on the robot enable the robot to capture, observe, and imitate

* Corresponding author.

E-mail address: chencj@qut.edu.cn (C. Chen).<https://doi.org/10.1016/j.rcim.2021.102167>

Received 23 October 2020; Received in revised form 19 February 2021; Accepted 23 March 2021

0736-5845/© 2021 Elsevier Ltd. All rights reserved.

the movements of the user. Unlike kinesthetic teaching and teleoperation, passive observation does not require a physical robot.

Teleoperation helps in handling dangerous, remote, or delicate tasks via robotic manipulators with enhanced safety at a lower cost and better accuracy. A teleoperation system includes a master subsystem, which is manipulated by an operator, and some slave subsystems that help handle the remote environment. In teleoperation, the perception of remote scenes and an effective human–computer interface are the key elements for realizing robot teleoperation. To enhance the human–computer interaction, we designed an augmented reality (AR)-based robot teleoperation system using RGB-D imaging and an attitude teaching device. The system provides a smooth human–robot interaction mode that enables a local operator to program the remote robot efficiently. The main contributions of this paper are the following:

- (1) An effective and inexpensive human–robot interaction method using RGB-D imaging and an attitude teaching device is proposed. With the human–robot interaction method, a local operator can complete the trajectory planning (including the path planning and orientation planning of the end effector of the robot) of a remote robot effectively.
- (2) A tele-AR registration method, which makes the coordinate system of the virtual robot coincide with that of the physical robot, is proposed.

The remainder of this paper is organized as follows: Section 2 summarizes related studies in this area. Section 3 describes the AR-based robot teleoperation teaching system and key related key technologies. Section 4 discussed the system design and user studies. Finally, section 5 concludes the paper.

2. Related work

In teleoperation demonstration, a user operates an input device to drive a remote robot. A teleoperation system typically includes a master subsystem, which is manipulated by an operator, and some slave subsystems that handle the remote environment [7]. Therefore, human–robot interaction is one of the key aspects of PbD for remote robots. Human–robot interaction, which can combine the skills of both humans and robots, has been widely studied. Tsarouchi et al. extensively reviewed human–robot interaction and its challenges in task planning and programming [8]. Using the concept of cyber–physical systems (CPS), Liu and Wang presented a remote human–robot collaboration system. This system can work in four different modes base on different scenarios. A remote robot control system and a model-driven display system were designed and tested in different scenarios. The test results showed that the human–robot collaboration system has considerable potential in hazardous manufacturing environments [9]. Nikolakis et al. designed a safe human-robot collaborative assembly system that can detect a human, and rapidly evaluate the distance between the human and the robot to avoid collision [10]. Currently, new types of human–computer interaction methods and devices are being incorporated for robot programming.

Haptic feedback is a human–computer interaction method whereby sensory feedback is provided through touch and force. It has been widely used in virtual reality (VR) and teleoperation applications [11]. Frazelle et al. designed a haptic continuum interface for the teleoperation of extensible continuum manipulators [12]. Ni et al. [13] proposed an intuitive user interface for programming welding robots remotely using AR with haptic feedback. The system was equipped with a depth camera

to reconstruct the surfaces of workpieces. A haptic input device allowed users to define the welding paths along these surfaces. The main challenge in haptic-based teleoperation is the real-time haptic rendering in the event of a network delay.

With the development of gesture devices, gesture-based technology has been used for teleoperation. For example, Du et al. presented a real-time remote robot teleoperation method using Kinect-based hand tracking. The hand pose was used as a model to specify the pose of a remote robot end-effector [14]. Tsarouchi et al. proposed a high level robot programming method that can move a robot in different directions by detecting body and hand gestures [15]. Jha et al. presented a learning scheme from a demonstration-based trajectory planner for an industrial robot end-effector to imitate human arm motions [16]. A gesture-based teleoperation system can imitate human motions. However, it cannot realize an accurate positioning and movement path planning.

To establish an accurate 3D model of the remote environment of a robot, a real-time 3D reconstruction technology has been used in teleoperation. Chen et al. proposed a VR and Kinect-based immersive teleoperation system and realized the reconstruction and simultaneous virtual environment creation of an unstructured agricultural scene [17]. Wang et al. presented a 3D model-driven remote robotic assembly system, which could construct 3D models at runtime to represent unknown geometries at the robot side. With the help of 3D models over the Internet, a remote operator can manipulate a real robot instantly to complete remote assembly operations [18]. A real-time 3D reconstruction, however, is time-consuming. Recently, depth sensors have been widely used in teleoperation. Kent et al. presented constrained positioning and point-and-click human–computer interface, which incorporates scene information from the depth data of remote robot environment into the grasp pose specification process. This reduces the number of 3D conversions inputted by the user [19]. Vakanski et al. integrated a visual servoing tracking control to robustly follow a trajectory generated from the observed sequence of images of demonstrations. The constraints originating from the use of a visual servoing controller are incorporated into the trajectory learning phase [20].

Electromyography (EMG) signals can reflect muscle activation. Surface electromyography (sEMG) is a noninvasive technique for detecting the electrical activity of muscles. It has been used in teleoperation to infer human motion intention. Luo et al. proposed a hybrid shared control method for a mobile robot with omnidirectional wheels. Operators utilized a 6-DOF haptic device and an EMG signal sensor to control the mobile robot [21]. Huang et al. developed a controller comprising a variable gain scheme to deal with fast-time-varying perturbations to enhance the tracking performance of a teleoperation system. The gain of the controller could be adjusted linearly based on EMG signals collected from a Myo wearable armband [22]. Park et al. proposed a path and impedance planning method for the impedance control of a robot based on PbD through telemanipulation using a surface EMG. The sEMG signals were used to estimate the impedance of the robot [23]. Goto et al. developed a hands-free remote operation system for a mobile robot using electrooculogram (EOG) and electromyogram (EMG). The motion commands generated by the processed EOG and EMG signals were used to drive the mobile robot [24]. Abibullaev et al. proposed a framework that integrates a brain–computer interface (BCI) and a humanoid robot to develop a brain-controlled telepresence system with multimodal control features. In this system, the low-level control is executed by PbD models, and the higher-level cognitive commands are produced by the BCI system [25].

With the development of VR and AR technologies, physical robots are being replaced by digital ones. Thus, the safety and effectiveness of

PbD can be enhanced. AR technology is typically used in remote maintenance. For example, Mourtzis et al. developed a cloud-based service-oriented AR system for remote maintenance. Using this system, on-site technicians and the manufacturer can communicate with each other through the cloud platform; thus, the manufacturer can provide the on-site technicians with AR instructions for maintenance [26]. AR can also help superimpose computer-generated information onto the physical world and has been proven to be effective in programming robots through demonstration [27–29]. Hedayati et al. explored how to leverage AR technology to improve robot teleoperation. Three aerial robot teleoperation interfaces using AR were presented, designed and evaluated. A participant user study showed that the teleoperation interfaces significantly improved the objective measures of teleoperation performance and speed while reducing crashes [30]. Walker et al. designed an AR-based teleoperation system to increase operation effectiveness. This system enables users to control a virtual robot surrogate that foreshadows physical robot actions, rather than directly operating the physical aerial robot [31]. Michalos *et al.* designed an AR tool for operators to provide multimodal support. In this system, the trajectory and workspace of the robot and other information of the production process can be visualized using AR technology so that the AR tool can enhance the operator's safety [32].

In AR-based programming of an industrial robot, the robot itself is virtually projected onto the physical industrial environment. The alignment of the virtual robot with the physical environment is achieved using different types of AR registration methods. Operators can test their programming paths by controlling the movements of virtual robots through their environment. Several AR-based industrial robot PbD systems have been developed. Chong *et al.* [33] discussed the use of AR environments to make robot programming more intuitive. They showed how AR could be used to move an industrial robot through a 3D environment without colliding with other objects. Fang *et al.* [34–37] presented an industrial robot PbD system with emphasis on trajectory planning (considering the dynamic constraints of the robot), orientation planning of the robot end-effector, human–virtual robot interaction methods, and adaptive path planning and optimization methods. Aleotti *et al.* [38] presented a visuo-haptic AR system to manipulate objects and learn tasks from demonstrations by humans. The system enabled users to operate a haptic device to interact with objects in a virtual environment. Chen *et al.* [39] presented an AR-based interactive robot teaching programming system. With this system, unskilled shopworkers can operate a handheld teaching device to make the end-effector of a virtual robot to follow the endpoint of the handheld teaching device. Thus, the path of the virtual robot can be planned or tested interactively. Vega *et al.* [40] presented a system to command and collaborate with robots. An AR interface was used to allow a user to send high-level requests to a robot,

in order to preview, approve, or modify the computed robot motions.

AR technology is widely used in the field of robot programming. However, research and industrial applications of AR-based teleoperation of robots are limited. In the AR-based teleoperation of a robot, the robot is at the remote side, whereas the operator is at the local side. Hence, the local operator cannot operate the remote robot cell manually. This increases the difficulty of AR registration, path planning, and task planning. This paper presents an AR-based robot teleoperation system that can capture the RGB-D images of remote environments. The local operator can define the trajectory of the robot endpoint using RGB-D imaging and an attitude teaching device. Subsequently, the defined trajectory can be tested using the AR technology.

3. AR-based robot teleoperation teaching system and key technology

To improve the flexibility and human–computer interaction of remote robot teaching programming and realize AR-based online teaching programming, a teaching programming for welding was taken as an example, and an AR-based industrial robot teleoperation teaching system using RGB-D imaging and an attitude teaching device was developed. As shown in Fig. 1, the remote side mainly includes a physical robot, a robot controller, and an RGB-D camera (Kinect camera), that captures images of the remote side environment and transmits them to the local side; the teaching operator operates the mouse and attitude teaching device at the local side to complete the teaching programming for robot teleoperation. The procedures for teaching programming are as follows:

- (1) The local side teaching operator is equipped with a mouse to select the path trajectory points to be reached by the robot end effector from color images, and the computer calculates the 3D coordinates of each path point in the robot coordinate system to form the working path of the robot end effector.
- (2) The attitude teaching device is used to plan the attitude of each position on the working path.
- (3) The AR registration technology is used to superimpose the virtual robot model on the color images of the physical robot working environment collected in real time, and the planned path and attitude are used to drive the movement of the virtual robot model in order to simulate and verify the feasibility of robot trajectory (path and attitude). Finally, the robot control program is generated and sent to the remote robot controller for execution.

Compared with conventional teaching methods, the proposed teaching system and method have several advantages. First, the robot

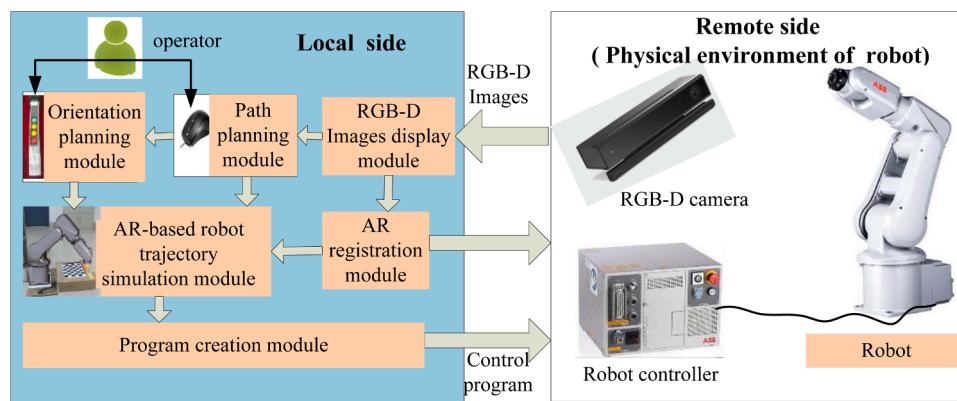


Fig. 1. Structure of the teaching system.

teaching programming can be achieved without modeling the environment of the remote side. Second, no manual operations, including the arrangements of AR registration cards and manual marking of the coordinates of the machined parts in the robot base coordinate system, are required at the remote side. In other words, the operator need not directly manipulate the physical robot. Therefore, the proposed teaching system is applicable to robot teleoperation programming and mobile robot programming.

To realize the above-mentioned AR-based industrial robot teleoperation methods using RGB-D imaging and attitude teaching device, two issues are resolved in this study. First, the AR registration is done online so that the base coordinate system of the virtual robot model coincides with that of the remote physical robot. Second, an advanced human–computer interaction is achieved so that the operator can define the trajectory (e.g., path, attitude, and motion speed) of the remote robot locally. The following section presents an AR registration method based on the positioning function of the remote robot.

3.1. AR registration method based on positioning function of remote robot

To verify the robot trajectory (path and attitude) in the AR environment, the virtual robot base coordinate system should coincide with the physical robot base coordinates in order to accurately superimpose the virtual robot model into the physical robot working environment, i. e., to achieve AR registration. In this teleoperation system, the physical robot base coordinate system is set as the world coordinate system, and by calculating the position and direction of the camera in the world coordinate system, the virtual robot model generated by the computer is accurately superimposed onto the images of the remote physical environment.

Common vision-based AR registration algorithms mainly include marker-based registration algorithms and markerless-based registration algorithms. The marker-based AR registration algorithms require

accurately placing markers in the robot working environment. The markerless-based registration algorithms also require marking the coordinate system in advance, which is evidently unsuitable for AR-based robot teleoperation programming. As in robot teleoperation programming, particularly online mobile robot programming, the remote side does not have an operator, and it is impossible to place registration cards and mark the coordinate system in advance.

In this study, an AR registration method based on the positioning function of remote robot is proposed. In this method, the marker is installed on the robot end effector, the 3D coordinates of the marker in the physical robot coordinate system can be obtained from robot control program. Then, the coordinates of the marker in the images can be obtained, and a coordinate conversion matrix M_{wc} between the robot base coordinate system and the color camera coordinate system is calculated.

At the local side, the matrix M_{wc} is used to accurately superimpose the virtual robot model into the physical environment, so as to realize the overlap between the virtual robot base coordinate system and the physical robot base coordinate system, and AR registration.

As shown in Fig. 2, the physical robot base coordinate system is $O_w - X_w Y_w Z_w$, and the color camera coordinate system is $O_c - X_c Y_c Z_c$. The steps involved in the proposed AR registration method are as follows:

- (1) The operator sends a control program through the local side teaching system to make the robot to replace the end effector with a registered marker (such as a small ball with a special shape and color). To improve the efficiency and accuracy of marker recognition, the marker used in this study is a small circular blue ball.
- (2) The operator sends the control program at the robot AR registration point to the remote side through the local side teaching system. The remote side physical robot controller receives the control program and controls the physical robot to realize point-to-point motion. In this study, the trajectory of the point-to-point

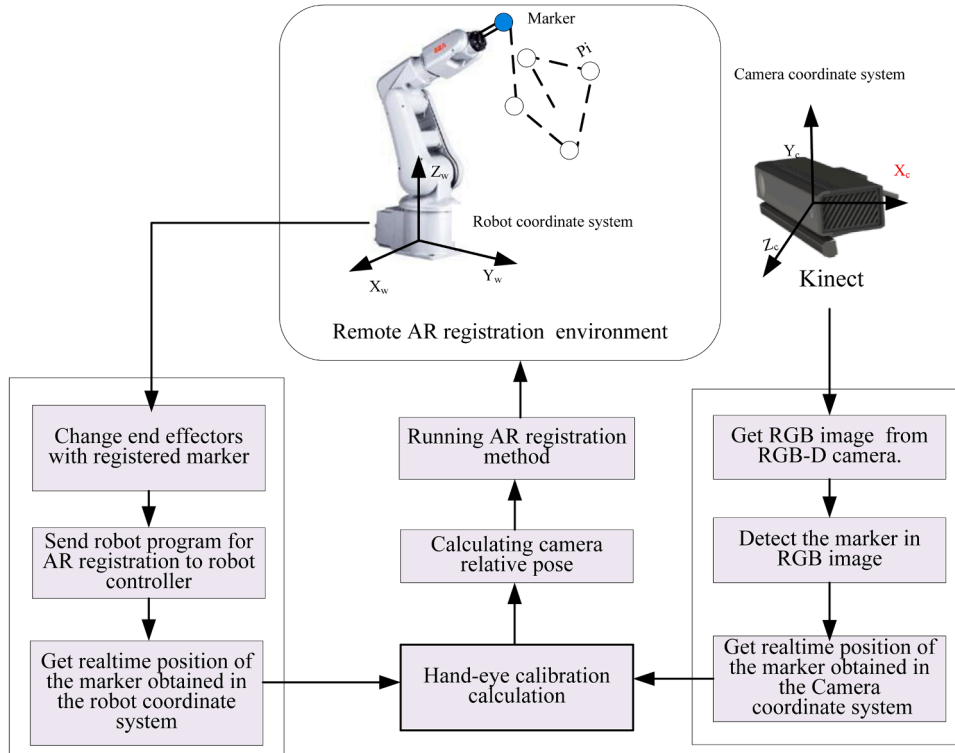


Fig. 2. Online AR registration.

motion is a square, and the intermittent motion points are the four end points of the square.

- (3) At each intermittent motion point, the local side teaching system reads the coordinates $P_i(x_{wi}, y_{wi}, z_{wi})$ of the registered marker in the physical robot base coordinate system from the remote side robot controller or control program, and collects the RGB-D image to identify the pixel coordinates $Z_i(u_i, v_i)$ of the feature points of the marker in the color images. Based on the imaging model of the Kinect cameras and camera parameters, the 3D coordinates $P'_i(x_{ci}, y_{ci}, z_{ci})$ of the marker feature points in the color camera coordinate system are calculated (the specific conversion method is described in Section 3.2). By obtaining the coordinates of four and more intermittent motion points in the robot base coordinate system and the color camera coordinate system, we can calculate the conversion matrix M_{wc} from the robot base coordinate system to the color camera coordinate system, which can be expressed as:

$$\begin{bmatrix} x_{ci} \\ y_{ci} \\ z_{ci} \\ 1 \end{bmatrix} = M_{wc} * \begin{bmatrix} x_{wi} \\ y_{wi} \\ z_{wi} \\ 1 \end{bmatrix}; M_{wc} = \begin{bmatrix} R & T \\ O & 1 \end{bmatrix}$$

where R is a 3×3 rotation matrix, and T is a 3×1 translation vector.

To improve the AR registration accuracy, the motion trajectory of robot can include multiple squares. By constructing and solving the least-squares problem, we can express M_{wc} as:

$$M_{wc} = \underset{R, T}{\operatorname{argmin}} \sum_{i=1}^n ||(RP'_i + T) - P_i||^2 \quad (1)$$

The minimum value of Eq. (1) is obtained by finding the solution where the derivative of R and T is 0 [41].

In summary, the proposed AR registration method for robot tele-operation meets the requirements of online industrial mobile AR teaching because it has the following advantages over other AR registration methods:

- (1) It does not require manual placing of any marker in advance, and the positioning function of the remote side robot can be used to realize AR registration in the online environment.
- (2) It is more suitable for multiple cameras. For example, three cameras are used in this study. The proposed AR registration

method can realize AR registration from the viewpoints of different cameras. In the teaching stage, the operator can switch to any of the three cameras as the main perspective for observing the virtual robot's working scene and path planning as required.

- (3) It is more suitable for mobile robots, in which the robot manipulator often moves from time to time after finishing a task.

3.2. Robot path position teaching based on RGB-D imaging

In robot programming, the robot motion trajectory is controlled by two parts: a motion path and a motion attitude. We take welding programming as an example to propose robot path position teaching methods based on RGB-D imaging. Using a mouse, an operator can select path points on the color images to obtain the pixel coordinates of the path points. Based on the camera external parameter matrix M_{wc} obtained from the Kinect camera imaging model and AR registration described in Section 3.1, the 3D coordinates of the path points in the physical robot coordinate system are calculated. The RGB-D images used for position teaching are teaching scenario images using the remote side main camera, the core of which is the conversion from the image coordinate system to the robot base coordinate system.

As shown in Fig. 3, the physical robot base coordinate system, Kinect color camera coordinate system, depth camera coordinate system, color image coordinate system, and depth image coordinate system are represented by $O_w - X_w Y_w Z_w$, $O_c - X_c Y_c Z_c$, $O_d - X_d Y_d Z_d$, and $O - U_c V_c$, $O - U_d V_d$, respectively. The classical pinhole camera model is used as the imaging model for the color and depth cameras [42]. Fig. 3 represents the conversion relationship between the point P and its projection point $p(u_d, v_d, d)$ on the imaging plane of the depth camera, u_d and v_d represent the coordinates of the pixels, and d represents the depth between the point P and the imaging plane of the depth sensor.

As shown in Fig. 3, the operator clicks on the color image, to obtain the path points of the robot. The pixel coordinates in the color image of the path point clicked by the operator are $p(u_c, v_c)$. Obtaining the 3D coordinates $P(x_w, y_w, z_w)$ of the path point in the robot coordinate system ($O_w - X_w Y_w Z_w$) involves the following steps:

Step 1: Obtain the pixel coordinates in the depth image from the color image pixel coordinates

Since the depth and color camera sensors on the Kinect device are relatively fixed, there is a pixel coordinate matching relationship between the color and depth images. We call the MapColor-FrameToDepthSpace function in the Kinect SDK to achieve a mapping

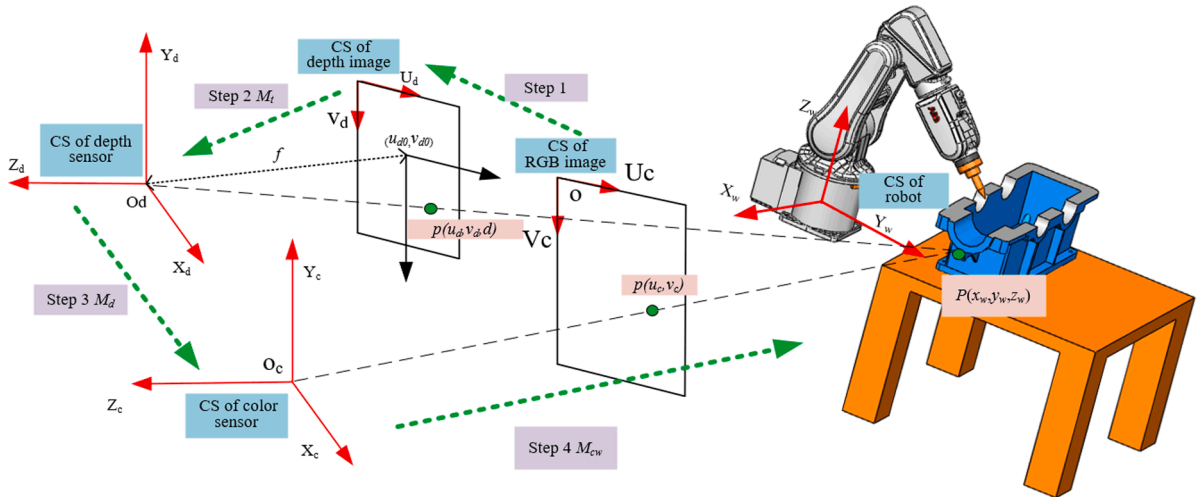


Fig. 3. Kinect camera imaging model and coordinate system conversion.

from the color image pixel coordinates to the depth image pixel coordinates. The color image pixel coordinates $p(u_c, v_c)$ and pixel depth d of the teaching path points are calculated. At this time, the pixels in the depth images can be denoted by $p(u_d, v_d, d)$.

$$\begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = W * \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} \quad (2)$$

Step 2: Calculate the coordinates in the depth camera coordinate system based on the internal parameter matrix M_t of the depth camera.

Based on the pinhole imaging model of the Kinect depth camera, for any point (x_d, y_d, z_d) in the depth sensor coordinate system $O_d - X_d Y_d Z_d$, the pixel coordinates (u_d, v_d) in the depth image coordinate system $O - U_d V_d$ can be calculated using the following equation:

$$d \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{dx} & 0 & u_{d0} & 0 \\ 0 & \frac{f}{dy} & v_{d0} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ z_d \\ 1 \end{bmatrix} = M_t \begin{bmatrix} x_d \\ y_d \\ z_d \\ 1 \end{bmatrix} \quad (3)$$

where M_t is the internal parameter matrix of the depth camera, d refers to the depth, u_{d0} and v_{d0} are the pixel coordinates of the center point on the depth image, f refers to the focal length of the depth camera, and d_x and d_y are the sizes of single pixel on the x-axis and y-axis of the depth sensor, respectively. From the above equation, we can express the 3D coordinates (x_d, y_d, z_d) of any point $p(u_d, v_d, d)$ on the depth image in the depth camera coordinate system as:

$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} \frac{(u_d - u_{d0}) * d * d_x}{f} \\ \frac{(v_d - v_{d0}) * d * d_y}{f} \\ d \end{bmatrix} \quad (4)$$

Step 3: Calculate the conversion matrix M_d from the depth camera coordinate system to the color camera coordinate system

Fig. 3 shows that the depth and color sensors have the same direction in the Kinect camera [43]. The origin of the depth camera coordinate system is shifted 51 mm in the positive X direction, which coincides with the origin of the Kinect color camera coordinate system. Therefore, the coordinate conversion matrix from the depth camera coordinate system $O_d - X_d Y_d Z_d$ to the color camera coordinate system can be expressed as:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -51 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_d \\ y_d \\ z_d \\ 1 \end{bmatrix} = M_d * \begin{bmatrix} x_d \\ y_d \\ z_d \\ 1 \end{bmatrix} \quad (5)$$

Combining Eqs. (2), (4), and (5) yields the conversion from the color image pixel coordinates to the RGB camera coordinates. In AR registration, after the RGB pixel coordinates of the marker feature point are recognized, the 3D coordinates in the color camera coordinate system can be obtained.

Step 4: Calculate the coordinates of the path points in the robot base coordinate system.

During the AR registration, presented in Section 3.1, the conversion matrix M_{wc} between the color camera coordinate system and the robot base coordinate system is obtained. Based on the camera imaging model, the conversion relationship between the coordinates (x_c, y_c, z_c) in the

color camera coordinate system and the coordinates (x_w, y_w, z_w) in the robot coordinate system at any point can be expressed as:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_{wc}^{-1} * \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (6)$$

Through the conversion between the coordinate systems of the color image pixel, depth image pixel, depth camera, color camera, and physical robot, the coordinates of the teaching path points clicked on the color image can be converted to the robot coordinate system, and the complete conversion can be expressed as:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = M_{wc}^{-1} * M_d * \begin{bmatrix} \frac{(u_d - u_{d0}) * d * d_x}{f} \\ \frac{(v_d - v_{d0}) * d * d_y}{f} \\ d \\ 1 \end{bmatrix}; \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = W * \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix} \quad (7)$$

3.3. Attitude and motion parameter teaching based on attitude teaching device

After completing the teaching of the robot path position point, it is also necessary to plan the attitude and motion parameters of the endpoint during the robot movement along the teaching trajectory.

The existing optical attitude tracking technology requires the installation of a camera. The system is large, lacks portability, and has a complicated calibration process. Therefore, it is unsuitable for robot teleoperation teaching. As shown in Fig. 4, a portable attitude teaching device based on the inertia sensor (IMU) is designed in this study. The portable device uses STM32F051K8U as the main control chip, measures the attitude of the teaching device in the geographic coordinate system through the BMI160 inertia sensor module, and transmits it to the local-side computers through a Bluetooth module. To control the motion parameters of the robot, the portable device is also equipped with four input buttons, i.e., controlling the start and stop of the attitude teaching and the acceleration and deceleration of the robot endpoint movement. The manipulator controls the attitudes of the portable attitude teaching device and the robot end effector. Therefore, it is important to establish a map between the attitude of the robot end effector and that of the attitude teaching device. In other words, the attitude of the attitude teaching device is the attitude of the robot end effector.

As shown in Fig. 4, the geographic coordinate system is recorded as $O_g - X_g Y_g Z_g$, the local coordinate system of the attitude teaching device is $O_a - X_a Y_a Z_a$, and the robot end effector coordinate system is $O_b - X_b Y_b Z_b$. The conversion between the geographical coordinate system and the robot base coordinate system is represented by a quaternion. In this system, as the robot base coordinate system has the same z-axis direction as the geographic coordinate system, and both meet the right-hand coordinate rule, the conversion relationship between the two coordinate systems is only related to the z-axis rotation angle. Therefore, when the robot is installed, the z-axis angle difference between the physical robot base coordinate system and the geographic coordinate system can be calibrated in advance, which can be expressed as $Q(Q.x, Q.y, Q.z, Q.w)$ using a quaternion.

The real-time attitude of the attitude teaching device in the geographic coordinate system is represented by a quaternion and denoted by q_g (e.g., IMU sensor output), the attitude of the attitude teaching device in the robot base coordinate system is denoted by q_w , and the conversion relationship between the geographical and robot base coordinate systems is denoted by Q . Accordingly, we have,

$$q_w = Q * q_g \quad (8)$$

After obtaining the attitude of the attitude teaching device in the

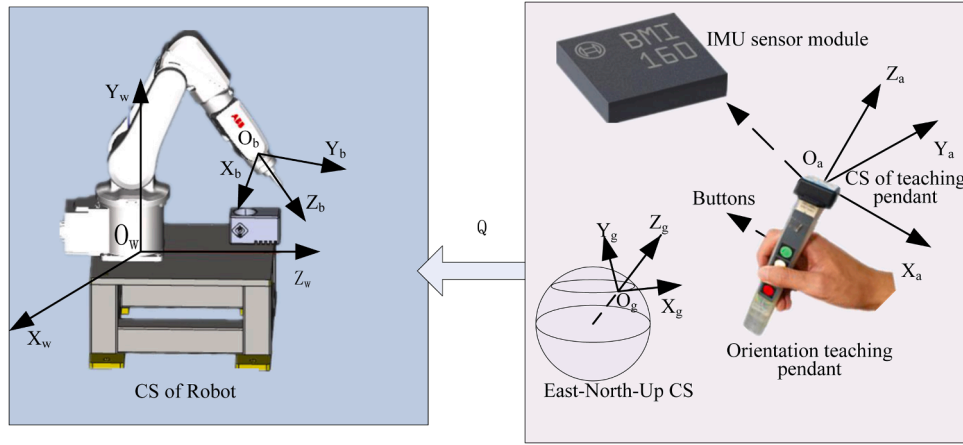


Fig. 4. Attitude teaching coordinate system.

robot base coordinate system from Eq. (8), we take the attitude as the attitude to be reached by the robot end effector. The quaternion attitude information represented by q_w is converted to the rotation matrix (represented by V) obtained from the Euler angles, which is expressed as

portable device to initialize the attitude of the end effector. The robot inverse kinematics module is used to solve the angle of each robot joint. The AR technology is used to superimpose the virtual robot model on the images of the physical environment.

$$V = \begin{bmatrix} 1 - 2q_w.y^2 - 2(q_w.z)^2 & 2(q_w.x * q_w.y) - 2(q_w.z * q_w.w) & 2(q_w.x * q_w.z) + 2(q_w.y * q_w.w) \\ 2(q_w.x * q_w.y) + 2(q_w.z * q_w.w) & 1 - 2q_w.x^2 - 2(q_w.z)^2 & 2(q_w.y * q_w.z) - 2(q_w.x * q_w.w) \\ 2(q_w.x * q_w.z) - 2(q_w.y * q_w.w) & 2(q_w.y * q_w.z) + 2(q_w.x * q_w.w) & 1 - 2q_w.x^2 - 2(q_w.y)^2 \end{bmatrix} \quad (9)$$

Since the position of the robot end effector is known, after planning the attitude of the point, the robot inverse kinematics can be used to solve the angle of each joint of the robot to drive the movement of the virtual robot model. Fig. 5 illustrates the procedures of attitude teaching using the portable attitude teaching device designed in this study.

We first initialize the portable attitude teaching device and establish a Bluetooth connection between the attitude teaching device and the local-side computers. When the operator presses the “start” button on the portable attitude teaching device to start teaching, the teaching system reads the starting point of the planning path to initialize the position of the robot’s endpoint, while receiving the attitude of the

During the teaching process, the local-side computers read the status of the portable attitude teaching device in real time. The teaching operator can control the speed of the virtual robot endpoint movement (in terms of the acceleration and deceleration) through the buttons on the portable attitude teaching device and the attitude of the virtual robot end effector through the attitude of the portable attitude teaching device. The robot inverse kinematics module is also used to solve the angle of each joint of the robot, and the virtual robot model is superimposed on the images of the physical environment using the AR technology. Thus, the operator can observe the teaching scenario in which the virtual robot and real world are combined, and control the motion speed and attitude in real time to complete the teaching. When the operator presses the “Stop” button, the teaching ends.

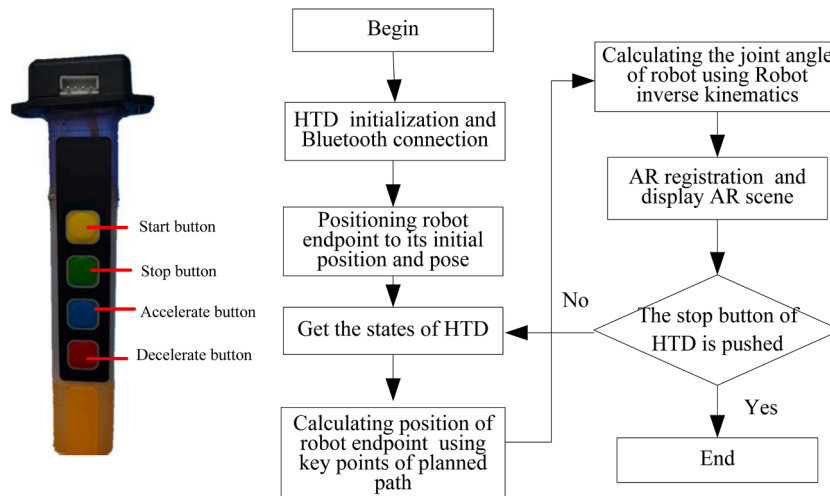


Fig. 5. Attitude teaching procedures.

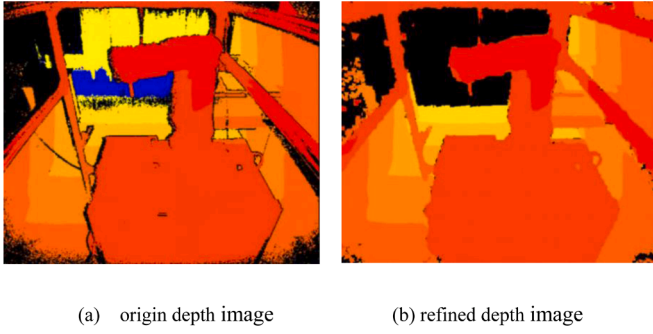


Fig. 6. Depth images refine.

3.4. Construction of remote scene based on KinectFusion API

In the proposed teaching system, there are two problems in obtaining the depth images through only one Kinect camera. First, the Kinect camera can only obtain the depth image from its own viewpoint, Hence it can only capture part of the remote scene. Second, as shown in Fig. 6 (a), the depth image contains a significant amount of noise. To improve the accuracy of the 3D coordinates of the teaching point in the robot coordinate system through the RGB-D imaging and ensure that the full

scene information of the remote robot working environment is obtained. In this teaching system, we place three Kinect cameras around the teaching scenario, obtain the 3D point cloud data through these cameras and stitch them, and generate full-scene depth images of the remote robot working environment from the stitched point cloud data. The specific implementation steps are as follows:

Three Kinect cameras are used to capture the color and depth images from the different angles of the remote robot working environment and the point cloud data are calculated using Eq. (4).

Eq (1) of the AR registration algorithm, presented in the previous section, is used to calculate the conversion relationship $M1_{wc} M2_{wc} M3_{wc}$ between the three Kinect cameras and the physical robot base coordinate system, i.e., the coordinate conversion relationship between the three Kinect cameras can be obtained. This conversion matrix transforms the data collected by the two Kinect cameras into the space of another Kinect. In other words, one of the main cameras is selected as the reference, and the point cloud data of the other two cameras are unified to the coordinate system of the reference camera.

The algorithm provided by KinectFusion API matches the 3D point cloud data to generate 3D point cloud data in the entire range of the teaching scenario, and the depth of the depth images is calculated from the point cloud data to generate depth images using the depth imaging model, as shown in Fig. 6(b). This is done to achieve accurate calculation of the depth and improve the path planning accuracy.

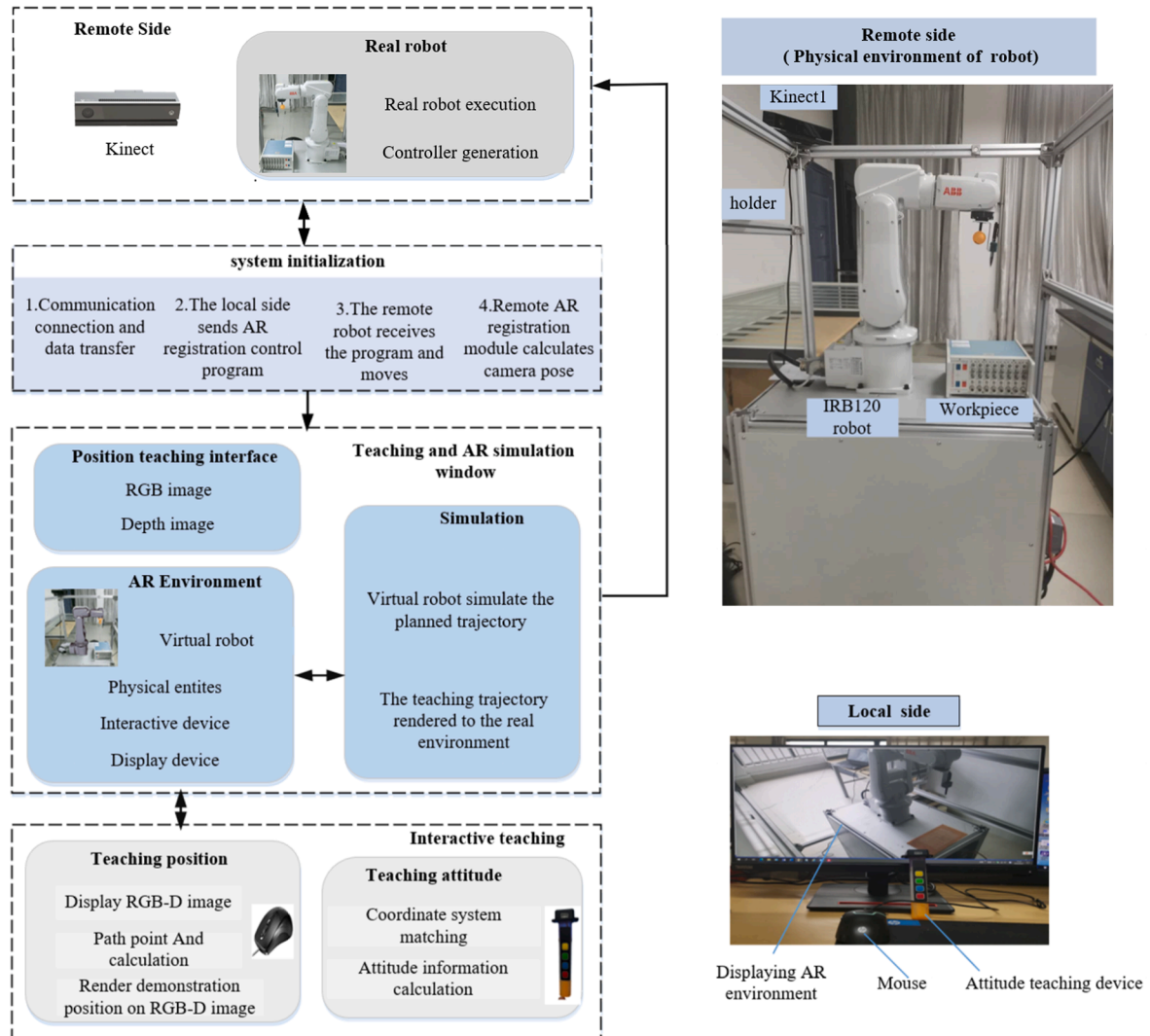


Fig. 7. Structure of AR-based robot teleoperation system.

4. System design and user studies

To verify the feasibility of the proposed method, we designed a 6-DOF AR-based robot teleoperation programming system. As shown in Fig. 7, the system is divided into two parts: a remote side and a local side. The former includes an ABB IRB120 robot, a teaching marker, three Kinect cameras, a bracket, and computers. The Kinect cameras are fixed onto the bracket and are aimed at the robot working area. The local side includes a computer (CPU: i7-7700HQ, RAM: 8 GB, graphics card: GTX1050Ti) and a portable attitude teaching device. The AR environment is composed of physical entities in the remote-side robot environment and virtual robot models. The computer mouse and portable attitude teaching device are used for the human-computer interaction. The AR environment is displayed on the screen of the local-side computers. Based on the VS2013 development environment, the software integrates ARToolkit 5.2 SDK, OpenSceneGraph3.0, OpenCV4.0, Kinect for Windows SDK2.0, and other interfaces. The remote-side system is only responsible for shooting the RGB-D image of the teaching scenario. It sends the images taken by it to the local side, receives the robot control program generated by the local side, and controls the robot movement. The local-side system is responsible for the entire teaching process and data processing.

4.1. System function

4.1.1. System initialization

When the teaching system starts to work, the system is first initialized, the online AR registration module is run, the camera pose in the robot base coordinate system is calculated, and the remote-side physical environment images are received. As shown in Fig. 8(a), based on the camera pose, the virtual robot model is superimposed on the physical environment to create an AR teaching environment. As shown in Figs. 8 (b) and (c), the color and depth images are displayed in the position teaching window for position point teaching. The red region in the depth map indicates the distance. The darker the color, the shorter is the distance. It is important to note that although the robot is fixed on a

platform in this experiment, the AR registration method proposed in this paper is also suitable for mobile robots that move after finishing a task.

4.1.2. Position teaching stage

As shown in Figs. 9(a) and (b), the operator uses the mouse to click the path points on the color images, and the clicked path points are displayed in the RGB-D image to help the operator perceive the physical environment and teaching points more clearly. The teaching system calculates the coordinates of the teaching points in the robot coordinate system using Eq. (7), to drive the virtual robot to reach the path points based on the default attitude. As shown in Fig. 9(c), the teaching path points and the attitude corresponding to the point will also be displayed on the color images via the AR teaching simulation window. *To make the teaching of the path points more accurate, the operator can make several attempts on the path points until the appropriate path points are selected, and switch to any of the three cameras as the main perspective for observing the robot's working scene and path planning, as required.*

4.1.3. Attitude teaching

After the position teaching is completed (as shown in Figs. 10(a)), the local-side computer reads the key points on the path in order and makes the endpoint of the virtual robot model to follow the teaching path. During the movement, the teaching attitude of the endpoint is calculated on the basis of the attitude of the attitude teaching device. Based on the current position and attitude of the robot endpoint, the robot's inverse kinematics model is used to calculate the angle of each joint axis of the robot to drive the movement of the virtual robot model, so as to realize the attitude planning during the movement of the virtual robot along the path. During attitude teaching, the teaching simulation interface is displayed in real time through the AR technology, and the operator adjusts the attitude of the robot endpoint based on the AR interface. Fig. 10 shows the motion map of the attitude teaching device that controls the movement of the robot end about the x-axis (Figs. 10(c) and (d)) and y-axis (Figs. 10(e) and (f)) from the initial state (Fig. 10(b)) in the attitude teaching process. During attitude teaching, the motion speed can be controlled by operating the buttons on the device.

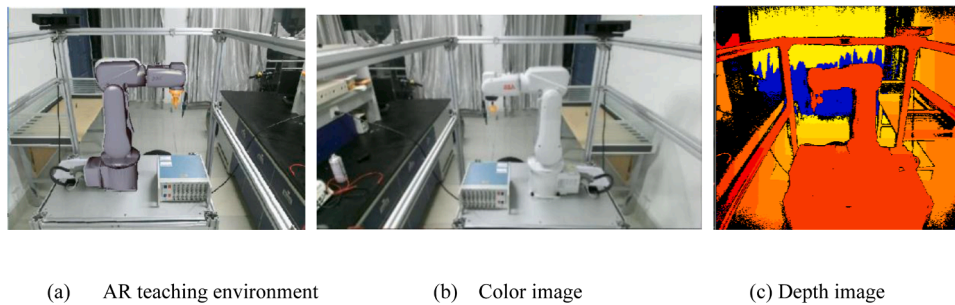


Fig. 8. System teaching window.

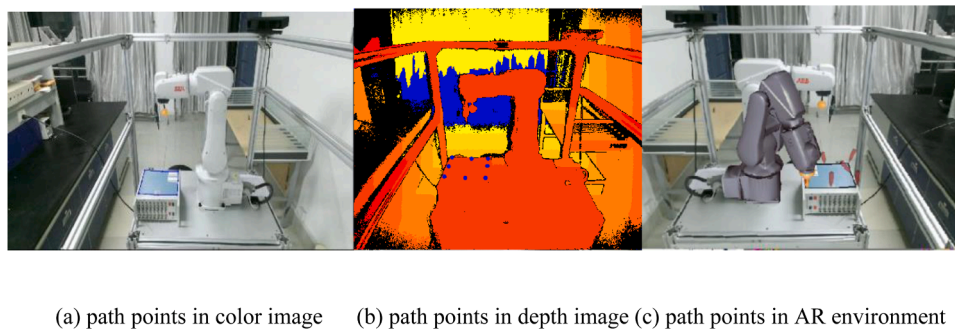


Fig. 9. Position teaching interface.

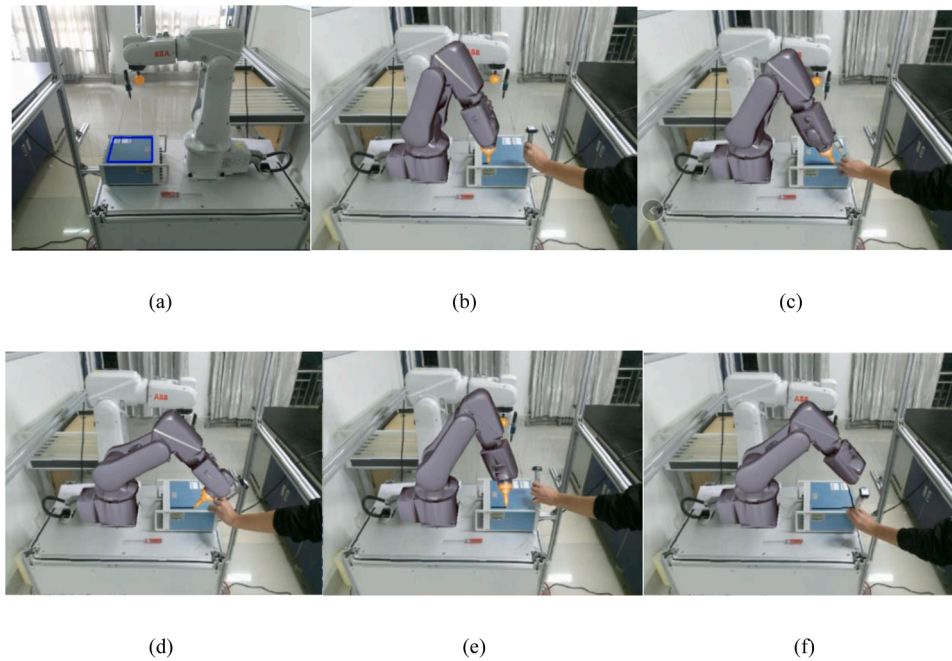


Fig. 10. Attitude teaching interface.

4.1.4. Trajectory simulation

After completing the trajectory planning, the teaching system drives the virtual robot model to reach the teaching attitude based on the rotation angle of each joint axis of the robot, so as to verify whether the teaching path is reasonable. If the trajectory is reasonable, the robot control program is generated on the basis of the rotation angle of each joint axis of the robot and is then sent to the remote side to control the movement of the physical robot (as shown in Fig. 11).

In this study, a non-mobile robot manipulator is taken as an example to demonstrate the functions and programming workflow of the proposed methods. However, the proposed methods are also suitable for mobile robot programming. In mobile robot manipulator programming over a wide working range, multiple RGB-D cameras should be used to capture RGB-D images of the working environment of the mobile robot. Once the position or orientation of the base under the robot manipulator changes, it is necessary to repeat the AR registration process to ensure that the virtual robot model generated by the computer can be accurately superimposed onto the images of the physical environment. Consequently, in this case the base of the robot manipulator cannot move when the robot manipulator is working. To apply the proposed programming method to a robot manipulator whose base can move while working, a 6 DOF real-time tracking system is required to track the real-time position and pose of the base.

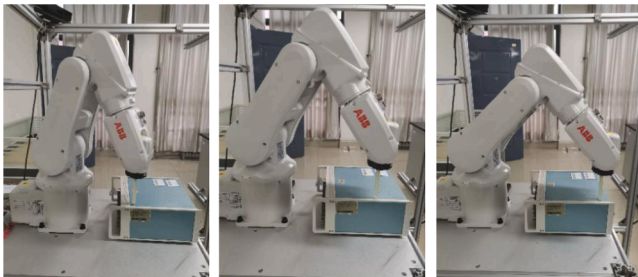


Fig. 11. Physical robot movement.

4.2. Trajectory accuracy

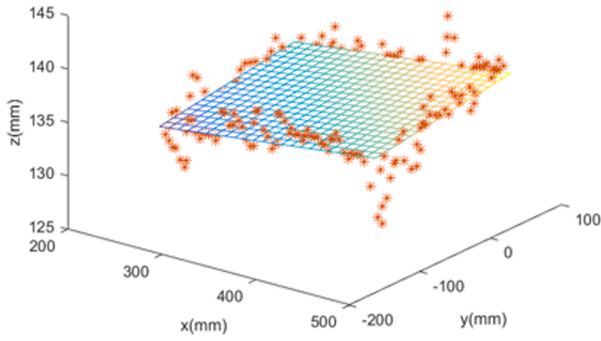
To verify the accuracy of the teaching path, the RGB and depth cameras of the Kinect setup were calibrated. By clicking on the four vertices of the rectangular plane on the color images using the mouse, the pixel coordinates of the four vertices could be interpolated (i.e., motion path in Fig. 9(c)). Eq (7) was used to calculate the 3D coordinates of all the pixel coordinates in the virtual robot coordinate system, so as to move the virtual robot model to the teaching position, save the 3D coordinates reached by the robot, process them in the MATLAB software, and use the least-squares method to fit all the path points that are inside the rectangular box. Fig. 12(a) shows the 3D coordinates calculated using the camera depth image data and the fitted plane. Fig. 12(b) shows the straight-line distance from all the path points to the fitted plane. The average error is 1.1871 mm, and the maximum error is 6.5941 mm. Fig. 12(c) shows the 3D coordinates and the fitted plane calculated from the depth data extracted after 3D reconstruction using KinectFusion API. As shown in Fig. 12(d), the average error of the 3D coordinate point to the fitted plane is 0.5050 mm, and the maximum error is 3.3261 mm.

The experimental results indicate that the 3D coordinates of the path points calculated by the depth images have evident fluctuation, and the average and maximum errors are high. This is mainly due to the Kinect sensor, measurement settings, and error in the depth data owing to the properties of the surface of the object. After filtering the depth data and extracting the depth map to calculate the 3D coordinates after point cloud fusion, the fluctuation in the 3D coordinate points is significantly reduced, and the error is decreased.

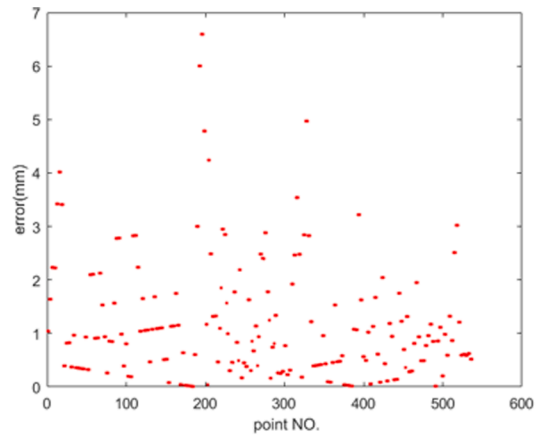
4.3. User survey

To evaluate the teaching usability of the teaching system, a user survey was conducted. The system test experiment was performed with ten volunteers, all aged between 22 and 30. Their task was to implement welding operation teaching for robots. None of the volunteers had any experience in robot teaching programming; hence, the testers were familiarized with the teaching system and offline programming software simultaneously before starting the experiment.

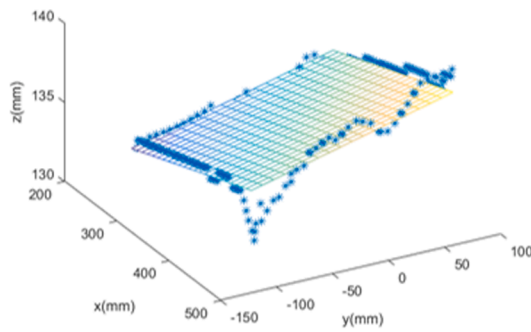
For the welding applications, the user was required to first operate



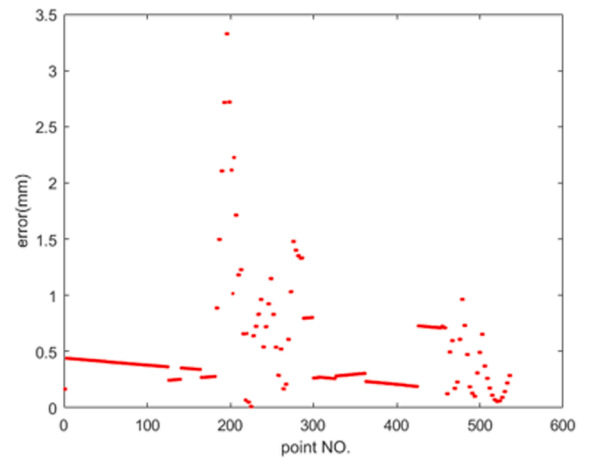
(a) Fitted plane by the camera depth



(b) Error of the camera depth



(c) Fitted plane by the optimized depth data



(d) Error of the optimized depth data

Fig. 12. Experimental accuracy verification.

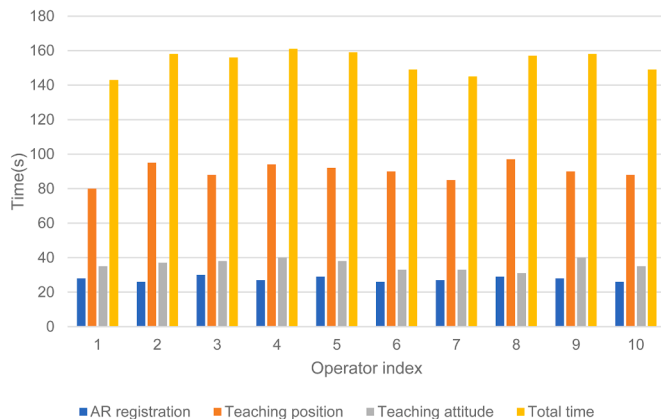


Fig. 13. Time taken by volunteers.

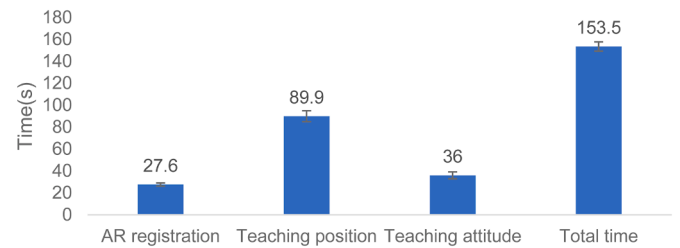


Fig. 14. Time to complete the teaching task.

the teaching system to complete the system initialization. As shown in Fig. 9(c), the key points on the welding path at a certain end of the robot are defined by selecting the edge of the welding workpiece with the mouse, and the end of the robot is moved on the basis of the attitude of the attitude teaching device.

To evaluate the performance of the teaching system in executing the

tasks, we tested the time taken by the volunteers to complete the AR registration, position teaching, and attitude teaching, and the total time to complete the teaching task. Fig. 13 shows the time taken by the ten volunteers to complete the task. Fig. 14 shows the average time taken by the volunteers to complete the teaching task. The system can complete the AR registration in a short time and meet the requirements of online AR registration. The position teaching takes a long time because the key points and attitude need to be considered. Inexperienced operators can complete robot teaching in a short time using this teaching system. This can be attributed to the fact that the teaching environment of the AR-based robot teleoperation teaching system is relatively intuitive and does not require modeling, and there is no requirement to calibrate the

Table 1
Questionnaire and score.

Questions	Score
a. The teaching system is easy to master	4.55
b. The augmented reality registration procedures of the system are simple.	4.64
c. The teaching interface of the teaching position in the system is easy to operate.	3.97
d. Portable attitude teaching device can help easily perform attitude teaching.	4.40
e. The AR teaching simulation window is very helpful in the teaching process.	4.57
f. The local side can well perceive the remote-side physical environment through the teaching window.	4.48
g. The probability of robot accidents is very low.	4.55
h. The overall teaching procedures are reasonable and can help complete basic robot motion teaching.	4.36

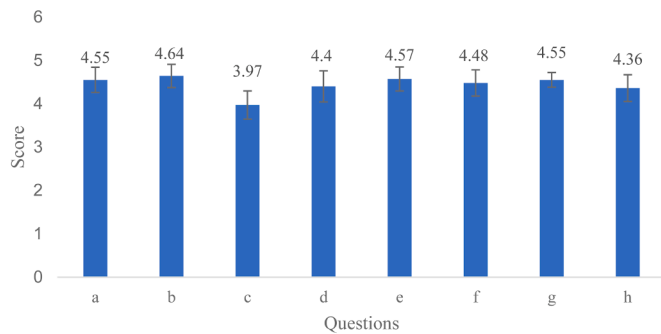


Fig. 15. Results of user survey.

coordinate system of the workpiece after the teaching is completed.

To evaluate a tester's experience for the system, we designed a questionnaire, as listed in Table 1, in terms of the operational safety, human–computer interaction, and rationality of the teaching procedures. After the tester completes the teaching task, a score is assigned on the basis of the questionnaire with an evaluation index of 1–5; the higher the score, the higher the degree of recognition [44].

Fig. 15 shows the test results of the ten volunteers, including the standard deviation. The results indicate that the testers who had no robot programming experience could easily master the teaching system after several trials and learning. They could successfully complete the AR registration, path teaching, and attitude teaching. However, the testers do not have a favorable opinion with regard to the operation of the position teaching interface of the system. Because of the large resolution of the color image in the teaching environment, it overlaps with the depth and AR images during the display process. The display positions of the images need to be manually adjusted; however, there is no need to adjust the images during attitude teaching. The testers are of the opinion that the robot attitude can be easily controlled using the attitude teaching device. In addition, the testers gave a high evaluation score for the safety of the teaching process and the operation steps. The volunteers confirm that the virtual robot's motion could be superimposed on the physical environment in real time through the AR technology for planning and verifying the teaching path, which can help them better understand the robot's motion state during the teaching process at the local side and complete the robot teaching to control the physical robot at the remote side.

5. Conclusions

An AR-based robot teleoperation teaching system using RGB-D imaging and attitude teaching device was developed in this study. In this system, AR teaching is realized by sending the images of the remote-side environment to the local side and superimposing the virtual robot model

onto the teaching environment images at the local side. The motion position of the virtual robot is determined by an operator who clicks on the color images, and the motion attitude is controlled using the attitude teaching device designed in this study. Through the simulation of the AR environment, the teaching pose could be displayed intuitively and in real time. The human–computer interaction was smooth, and the efficiency and operability of the teaching programming of the remote robot were improved.

A user survey showed that even people without programming experience could quickly master and use the system. The calibration process between the coordinate systems was also simplified. The teaching environment was displayed through three images, and the teaching process was divided into position teaching and attitude teaching, making the robot teaching process intuitive and easy to operate. The teaching process could be simulated in real time in the AR environment, so that operators can better observe and effectively control the robot's movement during the teaching process.

In the future, we intend to use the teleoperation programming system to perform more complicated tasks, such as intersecting line welding programming and mobile robot programming, to further test the ability of this system.

CRedit authorship contribution statement

Yong Pan: Methodology, Software, Writing – original draft. **Chengjun Chen:** Supervision, Conceptualization, Methodology, Writing – review & editing. **Dongnian Li:** Visualization, Software. **Zhengxu Zhao:** Supervision, Writing – review & editing. **Jun Hong:** Conceptualization.

Declaration of Competing Interest

None.

Acknowledgments

This work was co-supported by the National Natural Science Foundation of China (Grant No. 51475251) and Science & Technology Support Project for Young People in Colleges of Shandong Province (Grant No. 2019KJB020)

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.rcim.2021.102167](https://doi.org/10.1016/j.rcim.2021.102167).

References

- [1] H. Shin, S.H. Jung, Y.R. Choi, C. Kim, Development of a shared remote control robot for aerial work in nuclear power plants, *Nucl. Eng. Technol.* 50 (4) (2018) 613–618.
- [2] S.K. Ong, A.W.W. Yew, N.K. Thanigaivel, A.Y.C. Nee, Augmented reality-assisted robot programming system for industrial applications, *Robot. Comput. Integr. Manuf.* 61 (2020), 101820.
- [3] S. Calinon, Learning from demonstration (programming by demonstration), *Encyclopedia of Robotics* (2018) 1–8.
- [4] H. Ravichandar, A.S. Polydoros, S. Chernova, A. Billard, Recent advances in robot learning from demonstration, *Annual Review of Control, Robotics, and Autonomous Systems* (2020) 3.
- [5] S. Calinon, F. Guenter, A. Billard, On learning, representing, and generalizing a task in a humanoid robot, *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* 37 (2) (2007) 286–298.
- [6] R.A. Peters, C.L. Campbell, W.J. Bluthmann, E. Huber, Robonaut task learning through teleoperation, in: 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422) 2, IEEE, 2003, pp. 2806–2811.
- [7] M. Farahmandrad, S. Ganjefar, H.A. Talebi, M. Bayati, A novel cooperative teleoperation framework for nonlinear time-delayed single-master/multi-slave system, *Robotica* 38 (3) (2020) 475–492.
- [8] P. Tsarouchi, S. Makris, G. Chrysosolouris, Human–robot interaction review and challenges on task planning and programming, *Int. J. Comput. Integr. Manuf.* 29 (8) (2016) 916–931.

- [9] H. Liu, L. Wang, Remote human-robot collaboration: A cyber-physical system application for hazard manufacturing environment, *J. Manuf. Syst.* 54 (2020) 24–34.
- [10] N. Nikolakis, V. Maratos, S. Makris, A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace, *Robot. Comput. Integr. Manuf.* 56 (2019) 233–243.
- [11] M. Mihelj, J. Podobnik, *Haptics for virtual reality and teleoperation*, Vol. 67, Springer Science & Business Media, 2012.
- [12] C.G. Frazelle, A.D. Kapadia, I.D. Walker, A Haptic Continuum Interface for the Teleoperation of Extensible Continuum Manipulators, *IEEE Robot. Autom. Lett.* 5 (2) (2020) 1875–1882.
- [13] D. Ni, A.W.W. Yew, S.K. Ong, A.Y.C Nee, Haptic and visual augmented reality interface for programming welding robots, *Adv. Manuf.* 5 (3) (2017) 191–198.
- [14] G. Du, P. Zhang, J. Mai, Z. Li, Markerless kinect-based hand tracking for robot teleoperation, *Int. J. Adv. Robot. Syst.* 9 (2) (2012) 36.
- [15] P. Tsarouchi, A. Athanasatos, S. Makris, X. Chatzigeorgiou, G. Chryssolouris, High level robot programming using body and hand gestures, *Procedia CIRP* 55 (2016) 1–5.
- [16] A. Jha, S.S. Chiddarwar, Robot programming by demonstration using teleoperation through imitation, *Industrial Robot: An International Journal* (2017).
- [17] Y. Chen, B. Zhang, J. Zhou, K. Wang, Real-time 3D unstructured environment reconstruction utilizing VR and Kinect-based immersive teleoperation for agricultural field robots, *Comput. Electron. Agr.* 175 (2020), 105579.
- [18] L. Wang, A. Mohammed, M. Onori, Remote robotic assembly guided by 3D models linking to a real robot, *CIRP annals* 63 (1) (2014) 1–4.
- [19] D. Kent, C. Saldanha, S. Chernova, Leveraging depth data in remote robot teleoperation interfaces for general object manipulation, *Int. J. Rob. Res.* 39 (1) (2020) 39–53.
- [20] A. Vakanski, F. Janabi-Sharifi, I. Mantegh, An image-based trajectory planning approach for robust robot programming by demonstration, *Rob. Auton. Syst.* 98 (2017) 241–257.
- [21] J. Luo, Z. Lin, Y. Li, C. Yang, A teleoperation framework for mobile robots based on shared control, *IEEE Robot. Autom. Lett.* 5 (2) (2019) 377–384.
- [22] D. Huang, C. Yang, Z. Ju, S.L. Dai, Disturbance observer enhanced variable gain controller for robot teleoperation with motion capture using wearable armbands, *Auton. Robots* 44 (7) (2020) 1217–1231.
- [23] S. Park, W. Lee, W.K. Chung, K. Kim, Programming by demonstration using the teleimpedance control scheme: Verification by an semg-controlled ball-trapping robot, *IEEE Trans. Industr. Inform.* 15 (2) (2018) 998–1006.
- [24] S. Goto, O. Yano, Y. Matsuda, T. Sugi, N. Egashira, Development of Hands-Free Remote Operation System for a Mobile Robot Using EOG and EMG, *Electron. Commun. Jpn.* 100 (10) (2017) 38–47.
- [25] B. Abibullaev, A. Zollanvari, B. Saduanov, T. Alizadeh, Design and Optimization of a BCI-Driven Telepresence Robot Through Programming by Demonstration, *IEEE Access* 7 (2019) 111625–111636.
- [26] D. Mourtzis, V. Zogopoulos, E. Vlachou, Augmented reality application to support remote maintenance as a service in the robotics industry, *Procedia CIRP* 63 (2017) 46–51.
- [27] R. Bogue, The role of augmented reality in robotics, *Industrial Robot* (2020).
- [28] Z. Liu, W. Bu, J. Tan, Motion navigation for arc welding robots based on feature mapping in a simulation environment, *Robot. Comput. Integr. Manuf.* 26 (2) (2010) 137–144.
- [29] M.F. Zaeh, W. Vogl, Interactive laser-projection for programming industrial robots, in: 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality, IEEE, 2006, pp. 125–128.
- [30] H. Hedayati, M. Walker, D. Szafr, Improving collocated robot teleoperation with augmented reality, in: Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, 2018, pp. 78–86.
- [31] M.E. Walker, H. Hedayati, D. Szafr, Robot teleoperation with augmented reality virtual surrogates, in: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2019, pp. 202–210.
- [32] G. Michalos, P. Karagiannis, S. Makris, Ö. Tokçalar, G. Chryssolouris, Augmented reality (ar) applications for supporting human-robot interactive cooperation, *Procedia CIRP* 41 (2016) 370–375.
- [33] J.W.S. Chong, S. Ong, A.Y. Nee, K.B. Youcef-Youmi, Robot programming using augmented reality: An interactive method for planning collision-free paths, *Robot. Comput. Integr. Manuf.* 25 (3) (2009) 689–701.
- [34] H.C. Fang, S.K. Ong, A.Y.C Nee, Interactive robot trajectory planning and simulation using augmented reality, *Robot. Comput. Integr. Manuf.* 28 (2) (2012) 227–237.
- [35] H.C. Fang, S.K. Ong, A.Y.C Nee, Orientation planning of robot end-effector using augmented reality, *Int. J. Adv. Manuf. Tech.* 67 (9–12) (2013) 2033–2049.
- [36] H.C. Fang, S.K. Ong, A.Y.C Nee, A novel augmented reality-based interface for robot path planning, *Int. J. Interact. Des. Manuf. (IJIDeM)* 8 (1) (2014) 33–42.
- [37] H.C. Fang, S.K. Ong, A.Y.C Nee, Adaptive pass planning and optimization for robotic welding of complex joints, *Adv. Manuf.* 5 (2) (2017) 93–104.
- [38] J. Aleotti, G. Micconi, S. Caselli, Object interaction and task programming by demonstration in visuo-haptic augmented reality, *Multimed. Syst.* 22 (6) (2016) 675–691.
- [39] C. Chen, Y. Pan, D. Li, S. Zhang, Z. Zhao, J. Hong, A virtual-physical collision detection interface for AR-based interactive teaching of robot, *Robot. Comput. Integr. Manuf.* 64 (2020), 101948.
- [40] J.D. Hernández, S. Sobti, A. Sciola, M. Moll, L.E. Kavraki, Increasing Robot Autonomy via Motion Planning and an Augmented Reality Interface, *IEEE Robot. Autom. Lett.* 5 (2) (2020) 1017–1023.
- [41] O. Sorkine-Hornung, M. Rabinovich, Least-squares rigid motion using svd, *Computing* 1 (1) (2017) 1–5.
- [42] F. Flacco, T. Kroeger, A. De Luca, O. Khatib, A depth space approach for evaluating distance to objects, *J. Intell. Robot. Syst.* 80 (1) (2015) 7–22.
- [43] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, R. Siegwart, Kinect v2 for mobile robot navigation: Evaluation and modeling, in: 2015 International Conference on Advanced Robotics (ICAR), IEEE, 2015, pp. 388–394.
- [44] M. Ostanin, R. Yagfarov, A. Klimchik, Interactive Robots Control Using Mixed Reality, *IFAC-PapersOnLine* 52 (13) (2019) 695–700.