SCIENCE CHINA
Technological Sciences

• Article •

August 2023 Vol. 66 No. 8: 2406–2414
https://doi.org/10.1007/s11431-022-2205-9

Human-robot shared control system based on 3D point cloud and teleoperation

YANG ChenGuang[1*], ZHANG Ying[1], ZHAO GuanYi[1] & CHENG Long[2]

[1] *School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China;*
[2] *Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

Received June 22, 2022; accepted September 7, 2022; published online April 24, 2023

Owing to the constraints of unstructured environments, it is difficult to ensure safe, accurate, and smooth completion of tasks using autonomous robots. Moreover, for small-batch and customized tasks, autonomous operation requires path planning for each task, thus reducing efficiency. We propose a human-robot shared control system based on a 3D point cloud and teleoperation for a robot to assist human operators in the performance of dangerous and cumbersome tasks. The system leverages the operator's skills and experience to deal with emergencies and perform online error correction. In this framework, a depth camera acquires the 3D point cloud of the target object to automatically adjust the end-effector orientation. The operator controls the manipulator trajectory through a teleoperation device. The force exerted by the manipulator on the object is automatically adjusted by the robot, thus reducing the workload for the operator and improving the efficiency of task execution. In addition, hybrid force/motion control is used to decouple teleoperation from force control to ensure that force and position regulation will not interfere with each other. The proposed framework was validated using the ELITE robot to perform a force control scanning task.

teleoperation, 3D point cloud, human-robot shared control, hybrid force/motion control

Citation: Yang C G, Zhang Y, Zhao G Y, et al. Human-robot shared control system based on 3D point cloud and teleoperation. Sci China Tech Sci, 2023, 66: 2406–2414, https://doi.org/10.1007/s11431-022-2205-9

1 Introduction

Robots encounter difficulty in performing complex tasks autonomously in an unstructured environment [1]. With the development of artificial intelligence, robots are improving their capabilities. Hence, they will be expected to assume more responsibilities in multiple fields and become important assistants and tools for humans, with whom they will build interdependent relationships [2]. Human-robot cooperation allows robots to assist humans in the completion of complex and heavy work, and humans can adjust robot operation according to their practical needs. To reduce the workload to the human operator, we propose a system in which the end-effector orientation and force exerted on the object are automatically controlled by the robot, while the operator only controls the position of the end effector of a manipulator.

The concept of shared control was first proposed by Sheridan [3] as a control strategy in which a human operator and autonomous systems work in coordination to jointly control a robot remotely. The different degrees of freedom in a robot can be controlled manually or autonomously. Hence, the human operator and robot complete their own tasks, thus reducing the workload to the operator, improving the operation efficiency, and ensuring the safety, reliability, and stability of robot control. Humans and robots have complementary abilities, and their cooperation is beneficial in many cases. Robots are generally endowed with advanced sensors and strong computing power, allowing them to repeatedly perform high-load tasks with accuracy and without

*Corresponding author (email: cyang@ieee.org) experiencing fatigue with autonomy along a desired trajectory. On the other hand, humans have excellent cognitive ability, generally outperforming robots in situational perception and decision-making. They can provide robots with corrective actions, fine-tuning control, and situational guidance while performing tasks [4]. Effective human-robot cooperation can release humans from physically exhausting work, reduce the burden to operators, and improve the work efficiency and quality [5, 6]. By combining the strengths of humans and robots, great maneuverability can be achieved in unstructured environments. Human-robot shared control enables highly customized and small-batch manufacturing provided that the desired trajectory of the robot for each variation of the workpiece is efficiently and accurately determined [7]. Human-robot shared control is common in robot rehabilitation [8–10], search and rescue [11], remote operation [12], and many other fields. Remote operation has been widely used over the past few decades for tasks that are unfriendly to humans [13, 14]. Remotely operated robots can work in dynamic and unstructured environments and perform more diverse tasks than fully automated robots. Accordingly, we apply the concept of human-robot shared control through teleoperation while involving the operator in the control loop during robot operation.

We consider that the robot can automatically adjust the end-effector orientation when performing tasks based on a 3D point cloud. With the development of 3D imaging technology, various sensors (e.g., Kinect and RealSense) can acquire 3D data [15]. These sensors capture 3D data including shape and scale information [16, 17]. Collected 3D data have been applied in many fields, including unmanned driving [18], robotics [19, 20], medical therapy [21], and design [22]. On the other hand, 3D data can be represented in various forms, such as depth images and point clouds. A point cloud is a set of vectors in a 3D coordinate frame. These vectors are usually expressed in X-, Y-, and Z-axis coordinates and mainly represent the surface shape of an object. After acquiring 3D data through a depth camera, the coordinate information of a 3D point cloud can be processed. First, voxel filtering is used to downsample the point cloud and improve the computational efficiency [23]. Then, the region of interest is extracted through segmentation, and noise is removed by statistical filtering [23]. Finally, local surface fitting is applied to obtain the normal vector at any desired position in the 3D point cloud [24].

When a robot performs tasks that involve physical contact with the environment, such as grinding and assembly, motion control alone cannot ensure that the contact force during the task is maintained at the desired value, easily leading to task failure [25]. We aim for the manipulator to accurately track the desired trajectory and properly perform the intended task during interaction with the environment. Given the difficulty to obtain an accurate robot model, controllers based on neural networks have been proposed to optimize the tracking performance [26–28]. Ref. [29] combined admittance control with a force observer for a robot to exhibit a compliant behavior under the action of an external force. Ref. [30] proposed an adaptive impedance learning method to optimize the interaction between the environment and robot. Refs. [31, 32] used admittance control to deal with interaction problems in unknown environments. Ref. [33] proposed a robust control scheme to optimize the robot performance when interacting with environmental forces by defining a cost function involving trajectory error and force regulation. Ref. [34] proposed an adaptive admittance control scheme for robot interaction with the time-varying environment. A method for neural-network-based admittance control was proposed to achieve adequate interaction between the actuator and unknown environment considering a dead zone and force sensorless scheme in ref. [35]. Most of the abovementioned methods rely on wired sensors. To reduce costs and energy loss, wireless sensor networks seem promising for robotics [36].

Figure 1 shows the overall diagram of the proposed system, which is composed of modules for object modeling as 3D point clouds and hybrid force/motion control and teleoperation. In the figure, $\boldsymbol{N}_C$ is the normal vector at the desired position in the camera coordinate frame; $\boldsymbol{N}_R$ is the normal vector of the desired position in the robot base coordinate frame; ${}^{RB}_{C}\boldsymbol{R}$ is the rotation matrix from the camera to the robot base coordinate frame; ${}^{RE}_{RB}\boldsymbol{R}$ is the rotation matrix from the robot base to the end-effector coordinate frame; ${}^{RE}_{S}\boldsymbol{R}$ is the rotation matrix from the force sensor to the end-effector coordinate frame; $\Delta\boldsymbol{X}_T$ is the distance that the Touch X teleoperation device moves expressed in the end-effector coordinate frame; $\boldsymbol{F}_d$ is the desired force; $\boldsymbol{S}$ and $\boldsymbol{S}'$ are diagonal matrices whose diagonal elements are 0 or 1; $\Delta\boldsymbol{X}_{FR}$ is the sensor output for displacement adjustment; $\boldsymbol{F}_S$ is the force sensor measurement; $\Delta\boldsymbol{X}$ is the desired position increment of the end effector in the tip coordinate frame of the manipulator; $\boldsymbol{q}$ represents the desired joint angles, and $\boldsymbol{X}_d$ and $\boldsymbol{X}$ are the desired and actual positions of the end effector, respectively.

First, the 3D point cloud of the target object is obtained by a depth camera and preprocessed. The corresponding normal vector of the point cloud at the desired position is then obtained and recorded in the preprocessed point cloud. Teleoperation enables the remote control of the robot. By integrating the operator's skills and experience into the system, remote control is performed correctly and safely. To simultaneously control the motion and force of the end effector, a hybrid force/motion controller tracks the desired position ob-

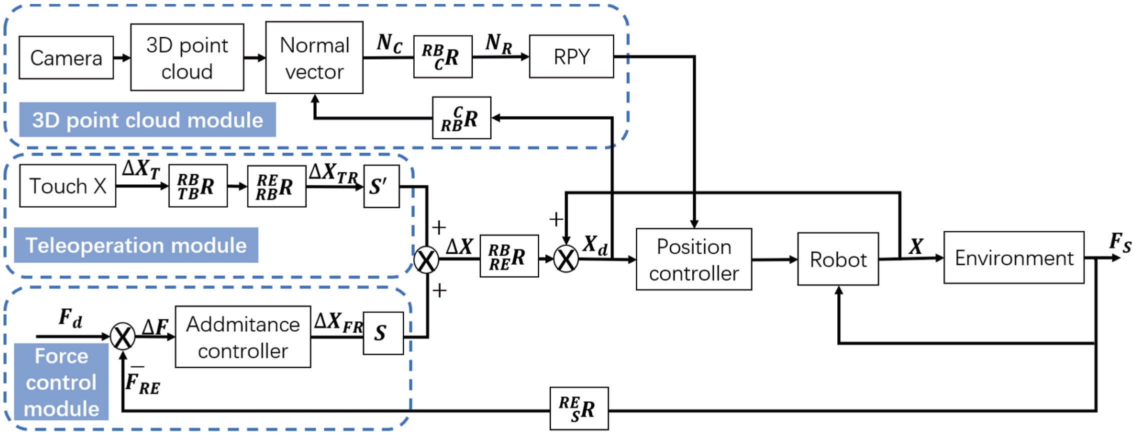

Figure 1 (Color online) Diagram of proposed control system.

tained from the teleoperation input and desired force set by the operator. The force controller and 3D point cloud acquisition of the target object greatly reduce the workload to the operator. In fact, the orientation and contact force of the end effector are automatically controlled by the robot, while the operator only needs to control the end-effector motion.

The major contributions of this study include the following three aspects.

(1) A human-robot shared control system based on 3D point clouds and teleoperation is developed to reduce the workload to the operator while ensuring safety.

(2) The end-effector orientation can be adjusted automatically based on the 3D point cloud of the target object.

(3) A hybrid force/motion controller decouples teleoperation and force control to avoid their interfering operation.

2 3D point cloud modeling

2.1 Eye-to-hand calibration

To represent the point cloud expressed in the camera coordinate frame in the robot base coordinate frame, the corresponding coordinate transformation is required when using a camera to generate the 3D point cloud of the target object. We apply eye-to-hand calibration [37] to the camera given its high accuracy and simplicity.

The diagram to perform eye-to-hand calibration is shown in Figure 2, whose process is mainly divided into the following three steps.

(1) Camera calibration. To obtain the coordinate transformation between the calibration plate and camera, we use the camera to identify the origin ($\boldsymbol{P}_{OC}$) of a checkerboard pattern and the coordinates of the camera coordinate frame corresponding to the unit points along the X ($\boldsymbol{P}_{XC}$) and Y ($\boldsymbol{P}_{YC}$) axes. According to these three coordinates, we can represent the X and Y axes of the checkerboard in the camera coordinate frame as $\boldsymbol{V}_{XC}$ and $\boldsymbol{V}_{YC}$, respectively. The representation of the checkerboard Z axis in the camera coordinate frame can be expressed by the cross product of the Y- and X-axis directions:

$$\boldsymbol{V}_{ZC} = \boldsymbol{V}_{YC} \times \boldsymbol{V}_{XC}. \tag{1}$$

Hence, we can obtain the rotation matrix from the checkerboard to the camera coordinate frame as follows:

$${}^{C}_{U}\boldsymbol{R} = \left[\, \boldsymbol{V}_{XC} \;\; \boldsymbol{V}_{YC} \;\; \boldsymbol{V}_{ZC} \,\right]. \tag{2}$$

In addition, the translation vector from the checkerboard to the camera coordinate frame is given by

$${}^{C}_{U}\boldsymbol{V} = \boldsymbol{P}_{OC}. \tag{3}$$

(2) Robot calibration. Similarly, to calibrate the robot coordinate frame, we should obtain the representation of the checkerboard with respect to the robot base coordinate frame. Dragging the end effector of the manipulator to the origin and unit length points of the X and Y axes, the current position of the tip of the manipulator is given by $\boldsymbol{P}_{OR}$, $\boldsymbol{P}_{XR}$, and $\boldsymbol{P}_{YR}$, respectively. These three coordinates provide the representation of the X and Y axes of the checkerboard with respect to the robot base

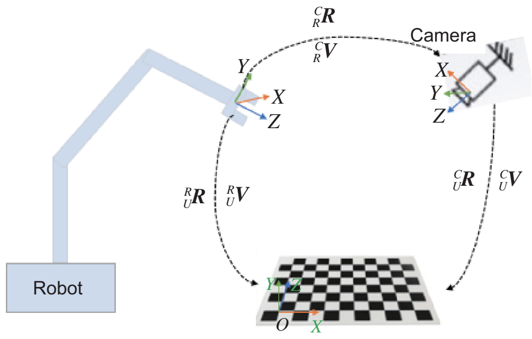
${}^{C}_{R}\boldsymbol{R}$
Camera
${}^{C}_{R}\boldsymbol{V}$
Y
X
Z
X
Y
Z
${}^{R}_{U}\boldsymbol{R}$
${}^{R}_{U}\boldsymbol{V}$
${}^{C}_{U}\boldsymbol{R}$
${}^{C}_{U}\boldsymbol{V}$
Robot
Y
Z
O
X

Figure 2 (Color online) Diagram of eye-to-hand calibration.

coordinate frame as $\boldsymbol{V}_{XR}$ and $\boldsymbol{V}_{YR}$, respectively. The representation of the Z axis in the robot coordinate frame can be expressed by the cross product of the Y- and X-axis directions:

$$\boldsymbol{V}_{ZR} = \boldsymbol{V}_{YR} \times \boldsymbol{V}_{XR}. \tag{4}$$

Hence, we obtain the rotation matrix from the checkerboard to the robot base coordinate frame as follows:

$${}^{R}_{U}\boldsymbol{R} = \left[\, \boldsymbol{V}_{XR} \;\; \boldsymbol{V}_{YR} \;\; \boldsymbol{V}_{ZR} \,\right]. \tag{5}$$

In addition, the translation vector from the checkerboard to the robot base coordinate frame is given by

$${}^{R}_{U}\boldsymbol{V} = \boldsymbol{P}_{OR}. \tag{6}$$

(3) Calibration results. After obtaining the rotation matrices and translation vectors from the checkerboard to the camera and to the robot base coordinate frames, we can obtain the rotation matrix and translation vector from the camera to the robot coordinate frame as follows:

$${}^{C}_{R}\boldsymbol{R} = {}^{U}_{R}\boldsymbol{R}{}^{C}_{U}\boldsymbol{R}, \tag{7}$$

$${}^{C}_{R}\boldsymbol{V} = {}^{U}_{R}\boldsymbol{V} + {}^{U}_{R}\boldsymbol{R}{}^{C}_{U}\boldsymbol{V}. \tag{8}$$

2.2 Point cloud preprocessing

To reduce the number of point clouds and accelerate software execution, we apply downsampling to the point clouds [23]. A 3D voxel grid is constructed, and the other points in each voxel are approximated with the barycenter of all points in the voxel. Hence, all the points in the voxel are represented by a center of gravity. The point cloud acquired by the camera contains objects unrelated to the task. To avoid the influence of those objects on the task and reduce the number of calculations, we extract the region of interest from the point cloud.

As noise remains in the point cloud, it is removed by statistical filtering [23], in which points that do not satisfy a target statistic are removed. Specifically, a sparse outlier is determined by calculating the distance distribution between every point and its neighbors. For each point, the average distance to all its neighbors is calculated. Assuming that the distance follows a Gaussian distribution, points with an average distance from their neighbors far from the mean of the Gaussian distribution can be defined as outliers, which are removed from the point cloud. Statistical filtering can be divided into three steps: (1) calculate the average distance from each point to its K neighbors; (2) calculate the sample mean and standard deviation of the K-neighborhood distance for the whole point set; (3) the average distance of the K neighbors of each point is compared with a distance threshold, and the points more distant than the threshold are marked as outliers and removed.

2.3 Normal vector calculation

We use local surface fitting to estimate the normal vector at each point [24]. Determining a surface normal at a point is similar to estimating a tangent surface normal. Specifically, we use the N points closest to point P to construct a plane and assume that the equation of the fitting plane is given by

$$\begin{aligned} & mx + ny + lz = k, \\ & m^2 + n^2 + l^2 = 1, \end{aligned} \tag{9}$$

where x, y, and z are 3D coordinate frames in the point cloud space, and m, n, l, and k are parameters of the plane equation. Therefore, solving plane equations can be expressed as solving for parameters m, n, l, and k. To obtain the best fitting plane, the sum of squares between all the points and the plane should be minimized:

$$\arg\min_{m,n,l,k} \sum_{j=1}^{N} |mx_j + ny_j + lz_j - k|^2. \tag{10}$$

We use the Lagrange multiplier method to solve this problem. First, the Lagrange function is constructed as follows:

$$F = \sum_{j=1}^{N} d_j^2 + \lambda\left(1 - m^2 - n^2 - l^2\right), \tag{11}$$

where $d_j = |mx_j + ny_j + lz_j - k|$ and $\lambda ! = 0$. Then, the partial derivatives of F with respect to m, n, l, and d_j are obtained as follows:

$$\frac{\partial F}{\partial d_j} = 2\sum_{j=1}^{N}\left(mx_j + ny_j + lz_j - k\right) = 0. \tag{12}$$

Next, it follows that

$$k = \frac{\sum_{j=1}^{N} x_j}{N}m + \frac{\sum_{j=1}^{N} y_j}{N}n + \frac{\sum_{j=1}^{N} z_j}{N}l. \tag{13}$$

Substituting k into d_j, we obtain

$$d_j = |m(x_j - x') + n(y_j - y') + l(z_j - z')|, \tag{14}$$

where $x' = \frac{\sum_{j=1}^{N} x_j}{N}$, $y' = \frac{\sum_{j=1}^{N} y_j}{N}$, and $z' = \frac{\sum_{j=1}^{N} z_j}{N}$. Let $\Delta x_j = x_j - x', \Delta y_j = y_j - y', \Delta z_j = z_j - z'$. Analogously, we can obtain

$$\begin{aligned} \frac{\partial F}{\partial m} &= 2\sum_{j=1}^{N}\left(m\Delta x_j + n\Delta y_j + l\Delta z_j\right)\Delta x_j - 2\lambda m, \\ \frac{\partial F}{\partial n} &= 2\sum_{j=1}^{N}\left(m\Delta x_j + n\Delta y_j + l\Delta z_j\right)\Delta y_j - 2\lambda n, \\ \frac{\partial F}{\partial l} &= 2\sum_{j=1}^{N}\left(m\Delta x_j + n\Delta y_j + l\Delta z_j\right)\Delta z_j - 2\lambda l. \end{aligned} \tag{15}$$

Then, we obtain

$$\boldsymbol{AX} = \lambda \boldsymbol{X}, \tag{16}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \sum \Delta x_j \Delta x_j & \sum \Delta x_j \Delta y_j & \sum \Delta x_j \Delta z_j \\ \sum \Delta x_j \Delta y_j & \sum \Delta y_j \Delta y_j & \sum \Delta y_j \Delta z_j \\ \sum \Delta z_j \Delta x_j & \sum \Delta z_j \Delta y_j & \sum \Delta z_j \Delta z_j \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} m & n & l \end{bmatrix}^{\mathrm{T}}. \tag{17}$$

Eq. (16) can be expressed as the problem of obtaining the eigenvalues and eigenvectors of matrix $\boldsymbol{A}$, which is the covariance matrix of the N nearest neighbor points. Let $\boldsymbol{X}$ be an eigenvector of this matrix. To minimize the sum of distances from the point to the plane, we choose the eigenvector corresponding to the point with the smallest eigenvalue as the optimal solution.

3 Teleoperation

For some scenarios beyond human reach and hazardous scenarios, such as fires and bomb disposals, operators should monitor and control remote robots to complete various operations and thus replace human intervention. Teleoperation maps the movement of a joystick onto the end effector such that a manipulator can move according to the human intention. Common mapping methods include scale mapping, position-velocity mapping, and workspace block mapping [38]. We map a leader-follower robot workspace through proportional mapping by setting appropriate coefficients.

We use Geomagic Touch X designed by SensAble Technologies as the leader device in the proposed system for robot control. Touch X is a haptic feedback device that includes hardware drivers and software packages (OpenHaptics Toolkit) [12]. The coordinate frames of the Touch X device and ELITE robot are shown in Figure 3, where TB is the base coordinate frame of Touch X and RB is the robot base coordinate frame. The following relation is defined:

$$\boldsymbol{\Delta X}_{TRB} = {}^{RB}_{TB}\boldsymbol{R} \begin{bmatrix} a_x & & \\ & a_y & \\ & & a_z \end{bmatrix} \begin{bmatrix} \Delta X(t) \\ \Delta Y(t) \\ \Delta Z(t) \end{bmatrix}, \tag{18}$$

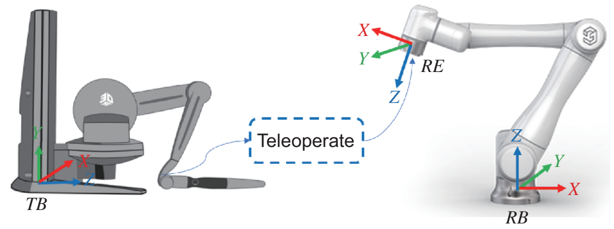

Figure 3 (Color online) Schematic diagram of teleoperation.

where $\Delta X(t) = X(t) - X(t-1)$, $\Delta Y(t) = Y(t) - Y(t-1)$, and $\Delta Z(t) = Z(t) - Z(t-1)$ are incremental distances of Touch X moving along the X, Y, and Z axes at time t, respectively; a_x, a_y, and a_z are the positive scalar mapping coefficients of Touch X to the ELITE robot along the corresponding axes; and $\boldsymbol{\Delta X}_{TRB}$ is the 3D increment in the manipulator movement.

4 Hybrid force/motion control

For the robot to properly interact with its environment, the force along the normal vector of the object should remain within the desired value. As force can be regulated by adjusting the position, if force control and position tracking are directly performed in the robot base coordinate frame, the X, Y, and Z axes would be simultaneously adjusted by two control variables. To precisely control force and perform teleoperation, these operations should be decoupled. To decouple force control from teleoperation, we implement the corresponding modules in the end-effector coordinate frame. The force controller regulates the motion of the manipulator along the Z axis of the tip coordinate frame, while Touch X controls the manipulator motion along the X and Y axes of the tip coordinate frame. First, we map the displacement increment along the X and Y axes of Touch X onto the end-effector coordinate frame as follows:

$$\Delta \boldsymbol{X}_{TRE} = {}^{RE}_{RB}\boldsymbol{R}^{RB}_{TB}\boldsymbol{R}\boldsymbol{\Delta X}_{T}, \tag{19}$$

where $\Delta \boldsymbol{X}_T$ is the Touch X displacement in the coordinate frame of Touch X, $\boldsymbol{\Delta X}_{TRE}$ is the Touch X displacement expressed as displacement in the manipulator tip coordinate frame, ${}^{RE}_{RB}\boldsymbol{R}$ is the rotation matrix from the base to the end-effector coordinate frame of the robot, and ${}^{RB}_{TB}\boldsymbol{R}$ is the rotation matrix from the Touch X base to the robot base coordinate frame.

Let the force measured by the force sensor be $\boldsymbol{F}_s$ with respect to the sensor coordinate frame. The force in the tip coordinate frame of the manipulator can be obtained by mapping $\boldsymbol{F}_s$ into the end-effector coordinate frame:

$$\boldsymbol{F}_{RE} = {}^{RE}_{S}\boldsymbol{R}\boldsymbol{F}_s, \tag{20}$$

where $\boldsymbol{F}_{RE}$ is the force expressed in the tip coordinate frame of the manipulator and ${}^{RE}_{S}\boldsymbol{R}$ is the rotation matrix from the force sensor to the end-effector coordinate frame. The tip of the manipulator is assumed to be perpendicular to the target object (e.g., human body). Thus, we only need to control the force along the Z axis in the tip coordinate frame of the manipulator to the desired value. Force control is formulated as follows:

$$\boldsymbol{M}\left(\ddot{\boldsymbol{X}}_d - \ddot{\boldsymbol{X}}_0\right) + \boldsymbol{D}\left(\dot{\boldsymbol{X}}_d - \dot{\boldsymbol{X}}_0\right) + \boldsymbol{K}\left(\boldsymbol{X}_d - \boldsymbol{X}_0\right)$$

$$= \boldsymbol{F}_{REd} - \boldsymbol{F}_{RE0}, \tag{21}$$

where $\boldsymbol{X}_0$, $\dot{\boldsymbol{X}}_0$, and $\ddot{\boldsymbol{X}}_0$ are the desired position, velocity, and acceleration of the end effector generated by force control, respectively; $\boldsymbol{X}_d$, $\dot{\boldsymbol{X}}_d$, and $\ddot{\boldsymbol{X}}_d$ are the virtual desired position, velocity, and acceleration generated by force errors, respectively; $\boldsymbol{F}_{REd}$ and $\boldsymbol{F}_{RE0}$ are the desired and actual forces, respectively; and $\boldsymbol{M}$, $\boldsymbol{D}$, and $\boldsymbol{K}$ are the controller parameters given by diagonal matrices with positive scalar elements along their diagonals. For simplicity and without loss of generality, we omit the acceleration term and obtain

$$\boldsymbol{D}\left(\dot{\boldsymbol{X}}_d - \dot{\boldsymbol{X}}_0\right) + \boldsymbol{K}\left(\boldsymbol{X}_d - \boldsymbol{X}_0\right) = \boldsymbol{F}_{REd} - \boldsymbol{F}_{RE0}. \tag{22}$$

Taking position error $\Delta\boldsymbol{X}_{FR} = \boldsymbol{X}_d - \boldsymbol{X}_0$, we have

$$\boldsymbol{D}\Delta\dot{\boldsymbol{X}}_{FR} + \boldsymbol{K}\boldsymbol{\Delta}\boldsymbol{X}_{FR} = \boldsymbol{F}_{REd} - \boldsymbol{F}_{RE0}. \tag{23}$$

Then,

$$(\Delta\dot{\boldsymbol{X}})_{FR} = \boldsymbol{D}^{-1}\left[(\boldsymbol{F}_{REd} - \boldsymbol{F}_{RE0}) - \boldsymbol{K}\Delta\boldsymbol{X}_{FR}\right], \tag{24}$$

$$\boldsymbol{\Delta X}_{FR}(t) = \boldsymbol{\Delta X}_{FR}(t-1) + \boldsymbol{\Delta}\dot{\boldsymbol{X}}_{FR}dt, \tag{25}$$

where dt is the control time interval and t denotes the timestep. Adding the control values of Touch X and force control, we obtain the final control value:

$$\boldsymbol{\Delta X} = \boldsymbol{\Delta X}_{TRE}\boldsymbol{S}' + \boldsymbol{\Delta X}_{FR}\boldsymbol{S}, \tag{26}$$

where

$$\boldsymbol{S}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \boldsymbol{S} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{27}$$

As $\boldsymbol{\Delta X}$ is based on the tip coordinate frame of the manipulator, we should represent it in the base coordinate frame of the robot for control:

$$\boldsymbol{X}_d = \boldsymbol{X}_{\text{cur}} + {}^{RB}_{RE}\boldsymbol{R}\boldsymbol{\Delta X}, \tag{28}$$

where $\boldsymbol{X}_{\text{cur}}$ is the current manipulator position.

The end posture of the manipulator is determined using the 3D point cloud. The next desired position of the manipulator, $\boldsymbol{X_d}$, is expressed in the camera coordinate frame, and the corresponding coordinate of the point in the camera coordinate frame is given by

$$\boldsymbol{X}_C = {}^{RB}_{C}\boldsymbol{R}^{-1}(\boldsymbol{X}_d - {}^{RB}_{C}\boldsymbol{V}). \tag{29}$$

By finding the point with minimum Euclidean distance between $\boldsymbol{X}_C$ and all points in the 3D point cloud, the corresponding normal vector of this point is regarded as the normal vector of desired point $\boldsymbol{N}_C$. As normal vector $\boldsymbol{N}_C$ is expressed in the camera coordinate frame, $\boldsymbol{N}_{RB} = {}^{RB}_{C}\boldsymbol{R}\boldsymbol{N}_C$ can be obtained by transformation to the robot base coordinate frame. After the expected normal vector is obtained, it should be converted into Euler angles and passed to the position controller:

$$\begin{aligned} &R = \arctan\left(-\boldsymbol{N}_{RB}[1], -\boldsymbol{N}_{RB}[2]\right), \\ &P = -\arcsin\left(-\boldsymbol{N}_{RB}[0]\right), \\ &Y = 0, \end{aligned} \tag{30}$$

where R, P, and Y are the rotation angles of the manipulator tip arm around the Z, Y, and X axes, respectively, starting from the base coordinate frame of the robot.

5 Experiments and results

The proposed system was validated through a mock B-scan ultrasonography experiment. Nowadays, the shortage of medical personnel and resources remains serious, and medical robots may be used to alleviate this problem. For instance, remote robot B-scan ultrasound technology allows to increase the examination efficiency, ensure the safety of sonographers, reduce their workload, and help patients to obtain timely treatment.

A sonographer should keep the probe with a proper contact force with the human skin for B-scan ultrasound examination and adjust the probe posture when necessary. Long-term handheld scanning may lead to muscle fatigue of sonographers, and the system wear and vibration may increase image variability. In addition, as a sonographer should be in close contact with patients, safety may not be guaranteed in some cases [39]. To ensure the safety of sonographers, they can be separated from the patients by leveraging teleoperation for examination. Using human-robot shared control for B-scan ultrasonography can reduce the workload to the sonographer. For demonstration, we acquired the 3D point cloud of a phantom, and the normal vector to each point in the 3D point cloud was calculated for automatically adjusting the direction of the probe along the normal to maintain the probe in the manipulator perpendicular to the phantom. In addition, the controller automatically adjusted the force exerted by the probe on the phantom to the desired value, while the sonographer only controlled the motion of the manipulator.

The experimental setup is shown in Figure 4. The COMATRIX camera was used to acquire the 3D point cloud of the phantom, and the ELITE robot performed the B-scan ultrasonography task on behalf of the sonographer. A force sensor was attached to the tip of the manipulator to measure the contact force, while Touch X enabled teleoperation and remote control of the robot. The coordinate frames of the robot base, end effector, force sensor, and Touch X were RB, RE, S, and

TB, respectively. The phantom was assumed to be stationary during the experiment.

The COMATRIX camera first acquired the 3D point cloud of the phantom. After calibration, the rotation matrix and translation vectors from the camera to the robot base coordinate frame were given by

$$\begin{aligned} {}^{RB}_{C}\boldsymbol{R} &= \begin{bmatrix} -0.07 & -0.997 & -0.038 \\ -0.99 & 0.075 & -0.093 \\ 0.095 & 0.031 & -0.993 \end{bmatrix}, \\ {}^{RB}_{C}\boldsymbol{V} &= \begin{bmatrix} -292.25 & 81.51 & 997.71 \end{bmatrix}. \end{aligned} \tag{31}$$

A color image of the phantom and its point cloud are shown in Figure 5. The original point cloud contained 1209738 points. Such many points increase the computational burden for normal vector calculation and point search. To reduce the number of computations, we applied voxel filtering and segmentation to the original point cloud, with a voxel size of 2 mm and the parameter for statistical filtering set to 80. The preprocessing results are shown in Figure 6. As noise polluted the original point cloud, we applied statistical filtering for removal, obtaining the results shown in Figure 6.

The number of points after preprocessing the point cloud was 49714, being only 4.1% of the original point cloud and greatly reducing the amount of data and calculations. We then applied local surface fitting to the resulting 3D point cloud to calculate the normal vectors to target points. All point positions and their corresponding normal directions were stored in a lookup table. The operator remotely controlled the movement of the manipulator tip through the Touch X haptic feed-

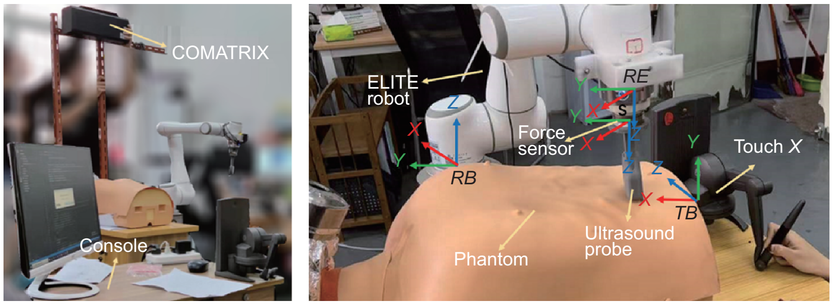

Figure 4 (Color online) COMATRIX camera (left) used to capture phantom 3D point cloud. Experimental equipment (right) for B-scan ultrasonography.

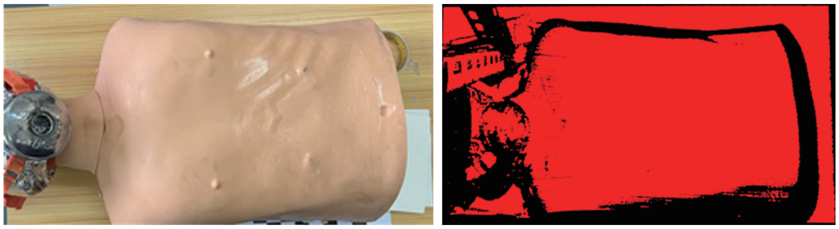

Figure 5 (Color online) Color image and 3D point cloud of phantom.

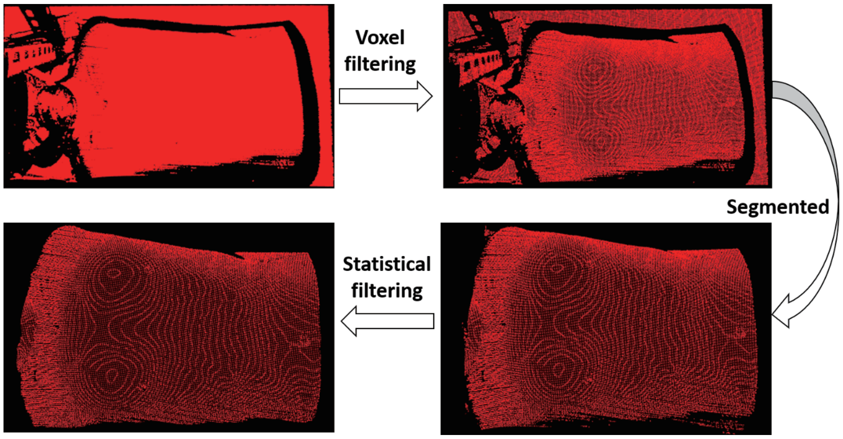

Figure 6 (Color online) Point cloud data processing process.

back device. The displacements of Touch X along the X and Y axes were mapped onto movements along the corresponding axes in the tip coordinate frame. The motion along the Z axis in the tip coordinate frame was regulated by force control. By adjusting the position along the Z axis, the force at the tip was kept at the desired value. The orientation of the end effector was automatically determined by the corresponding controller according to the normal direction stored in the lookup table.

The effectiveness of the proposed system was verified by using Touch X to control the manipulator to scan back and forth on the surface of the phantom. During scanning, the desired force was set to 5 N, and the scaling factor between the movements of Touch X and the manipulator was set to 1, as shown in Figure 7. During scanning, the position errors between the desired input position of Touch X and actual position of the end effector along the X and Y axes in the robot base coordinate frame were obtained as shown in Figure 8. The force error along the Z axis in the end effector frame is shown in Figure 9. The maximum motion tracking error was approximately 6 mm, and the maximum force tracking error was approximately 2 N, being acceptable for this task.

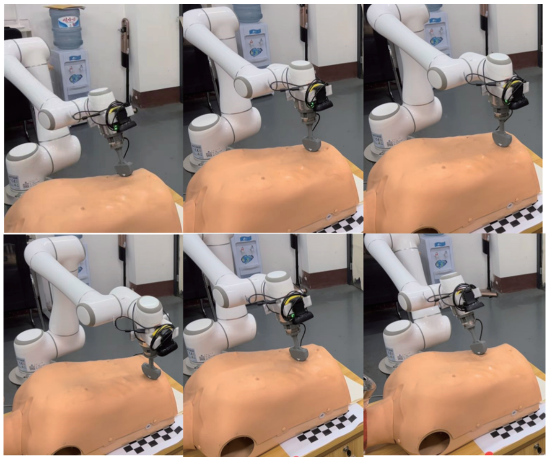

Figure 7 (Color online) Position and orientation of ultrasound probe over time during scanning.

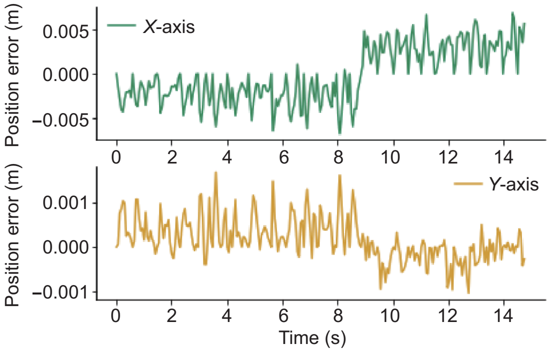

Figure 8 (Color online) Position error along X and Y axes of end effector with respect to robot base frame during B-scan ultrasonography task.

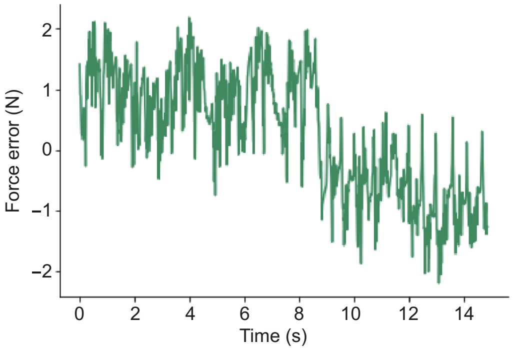

Figure 9 (Color online) Force error along Z axis in end effector with respect to robot base frame during ultrasound scanning.

6 Conclusions

We introduce a shared control system based on a 3D point cloud and teleoperation. The system leverages human skills and experience through teleoperation. Human-robot cooperation allows the robot to work according to the operator's intention while separating the operator from the working environment to ensure safety via teleoperation. For diverse tasks, such as polishing and ultrasound scanning, the end effector should remain perpendicular to the object surface while exerting appropriate force on the object during operation. We use a depth camera to obtain the 3D point cloud of the target object and then obtain the normal vector at any desired position on the surface. In addition, hybrid force/motion control allows the manipulator to adjust both the orientation of the end effector and force exerted on the object. This system aims to reduce the workload to the operator, who only needs to control the position of the manipulator tip, likely improving the efficiency and quality of task execution.

Given that ultrasonography may be interfered by bones and other structures, a clear image may not be obtained if keeping the probe strictly perpendicular to the body surface. Therefore, additional interfaces may be developed in future work for the sonographer to fine-tune the operation based on the existing vertical posture. In addition, considering the deformation when the probe touches the body, we will implement real-time planning of the end posture of the manipulator throughout scanning.

This work was supported by the National Natural Science Foundation of China (NSFC) (Grant No. U20A20200), the Major Research (Grant No. 92148204), the Guangdong Basic and Applied Basic Research Foundation (Grant Nos. 2019B1515120076 and 2020B1515120054), and the Industrial Key Technologies R&D Program of Foshan (Grant Nos. 2020001006308 and 2020001006496).

1 Yang C, Luo J, Liu C, et al. Haptics electromyography perception and learning enhanced intelligence for teleoperated robot. IEEE Trans Automat Sci Eng, 2019, 16: 1512–1521

2 Yang C, Luo J, Pan Y, et al. Personalized variable gain control with tremor attenuation for robot teleoperation. IEEE Trans Syst Man Cybern Syst, 2018, 48: 1759–1770

3 Sheridan T B. Space teleoperation through time delay: Review and prognosis. IEEE Trans Robot Automat, 1993, 9: 592–606

4 Li Y, Tee K P, Ge S S, et al. Building Companionship Through Human-Robot Collaboration. Cham: Springer International Publishing, 2013. 1–7

5 Kong K, Bae J, Tomizuka M. A compact rotary series elastic actuator for human assistive systems. IEEE ASME Trans Mechatron, 2012, 17: 288–297

6 Daniel R. Telerobotics, automation and human supervisory control. Automatica, 1994, 30: 1072–1073

7 Li Y, Tee K P, Chan W L, et al. Continuous role adaptation for human-robot shared control. IEEE Trans Robot, 2015, 31: 672–681

8 Wang J, Wang W, Hou Z G. Toward improving engagement in neural rehabilitation: Attention enhancement based on brain-computer interface and audiovisual feedback. IEEE Trans Cogn Dev Syst, 2020, 12: 787–796

9 Wang C, Peng L, Hou Z G, et al. Quantitative assessment of upper-limb motor function for post-stroke rehabilitation based on motor synergy analysis and multi-modality fusion. IEEE Trans Neural Syst Rehabil Eng, 2020, 28: 943–952

10 Luo L, Peng L, Wang C, et al. A greedy assist-as-needed controller for upper limb rehabilitation. IEEE Trans Neural Netw Learn Syst, 2019, 30: 3433–3443

11 Liu Y, Nejat G. Robotic urban search and rescue: A survey from the control perspective. J Intell Robot Syst, 2013, 72: 147–165

12 Hirche S, Buss M. Human-oriented control for haptic teleoperation. Proc IEEE, 2012, 100: 623–647

13 Yang C, Wang X, Cheng L, et al. Neural-learning-based telerobot control with guaranteed performance. IEEE Trans Cybern, 2017, 47: 3148–3159

14 Yang C, Wang X, Li Z, et al. Teleoperation control based on combination of wave variable and neural networks. IEEE Trans Syst Man Cybern: Syst, 2017, 47: 2125–2136

15 Liang Z, Guo Y, Feng Y, et al. Stereo matching using multi-level cost volume and multi-scale feature constancy. IEEE Trans Pattern Anal Mach Intell, 2021, 43: 300–315

16 Guo Y, Sohel F, Bennamoun M, et al. Rotational projection statistics for 3D local surface description and object recognition. Int J Comput Vis, 2013, 105: 63–86

17 Guo Y, Bennamoun M, Sohel F, et al. 3D object recognition in cluttered scenes with local surface features: A survey. IEEE Trans Pattern Anal Machine Intell, 2014, 36: 2270–2287

18 Li M, Zhang Y. 3D point cloud labeling tool for driving automatically. In: Proceedings of the 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). Auckland, 2020. 1666–1672

19 Chen X, Chen C H. Model-based point cloud alignment with principle component analysis for robot welding. In: Proceedings of the 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS). Taipei, 2017. 83–87

20 Lin H I, Cong M N. Inference of 6-DOF robot grasps using point cloud data. In: Proceedings of the 2019 19th International Conference on Control, Automation and Systems (ICCAS). Jeju, 2019. 944–948

21 Xiao D, Lian C, Deng H, et al. Estimating reference bony shape models for orthognathic surgical planning using 3D point-cloud deep learning. IEEE J Biomed Health Inform, 2021, 25: 2958–2966

22 Chen X, Ma H, Wan J, et al. Multi-view 3D object detection network for autonomous driving. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Hawaii, 2017. 6526–6534

23 Rusu R B, Cousins S. 3D is here: Point cloud library (PCL). In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation. Shanghai, 2011. 1–4

24 Hsieh C T. An efficient development of 3D surface registration by point cloud library (PCL). In: Proceedings of the 2012 International Symposium on Intelligent Signal Processing and Communications Systems. Tamsui, 2012. 729–734

25 Wang N, Chen C, Nuovo A D. A framework of hybrid force/motion skills learning for robots. IEEE Trans Cogn Dev Syst, 2021, 13: 162–170

26 Kong L, He W, Yang C, et al. Robust neurooptimal control for a robot via adaptive dynamic programming. IEEE Trans Neural Netw Learn Syst, 2021, 32: 2584–2594

27 He W, Chen Y, Yin Z. Adaptive neural network control of an uncertain robot with full-state constraints. IEEE Trans Cybern, 2016, 46: 620–629

28 He W, David A O, Yin Z, et al. Neural network control of a robotic manipulator with input deadzone and output constraint. IEEE Trans Syst Man Cybern: Syst, 2016, 46: 759–770

29 Yang C, Peng G, Cheng L, et al. Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks. IEEE Trans Syst Man Cybern: Syst, 2021, 51: 3282–3292

30 Huang H, Yang C, Chen C L P. Optimal robot-environment interaction under broad fuzzy neural adaptive control. IEEE Trans Cybern, 2021, 51: 3824–3835

31 Peng G, Yang C, He W, et al. Force sensorless admittance control with neural learning for robots with actuator saturation. IEEE Trans Ind Electron, 2020, 67: 3138–3148

32 Yang C, Peng G, Li Y, et al. Neural networks enhanced adaptive admittance control of optimized robot-environment interaction. IEEE Trans Cybern, 2019, 49: 2568–2579

33 Peng G, Chen C P, Yang C. Robust admittance control of optimized robot-environment interaction using reference adaptation. IEEE Trans Neural Networks Learn Syst, 2022, doi: 10.1109/TNNLS.2021.3131261

34 Peng G, Chen C L P, Yang C. Neural networks enhanced optimal admittance control of robot-environment interaction using reinforcement learning. IEEE Trans Neural Networks Learn Syst, 2022, 33: 4551–4561

35 Peng G, Chen C L P, He W, et al. Neural-learning-based force sensorless admittance control for robots with input deadzone. IEEE Trans Ind Electron, 2021, 68: 5184–5196

36 Lu Z, Wang N, Yang C. A novel iterative identification based on the optimised topology for common state monitoring in wireless sensor networks. Int J Syst Sci, 2022, 53: 25–39

37 Tsai R Y, Lenz R K. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. IEEE Trans Robot Automat, 1989, 5: 345–358

38 Conti F, Khatib O. Spanning large workspaces using small haptic devices. In: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference, Pisa, 2005. 183–188

39 Conti F, Park J, Khatib O. Interface design and control strategies for a robot assisted ultrasonic examination system. Springer Tracts Adv Robot, 2014, 79: 97–113