Robotica (2017) volume 35, pp. 1958–1974. © Cambridge University Press 2016
doi:10.1017/S0263574716000631

3D-point-cloud registration and real-world dynamic modelling-based virtual environment building method for teleoperation

Dejing Ni, Aiguo Song*, Xiaonong Xu, Huijun Li, Chengcheng Zhu and Hong Zeng

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, P.R. China. E-mails: ndj_seu@163.com, seuxxn@gmail.com, lihuijun@seu.edu.cn, zccins_seu@seu.edu.cn, hzeng@seu.edu.cn

(Accepted August 6, 2016. First published online: September 9, 2016)

SUMMARY
It is a challenging task for a human operator to manipulate a robot from a remote distance, especially in an unknown environment. Excellent teleoperation provides the human operator with a sense of telepresence, mainly including real-world vision, haptic perception, etc. This paper presents a novel virtual environment building method using the red–green–blue (RGB) colour information, the surface normal feature-based 3D-point-cloud registration method and the weighted sliding-average least-square-method-based real-world dynamic modelling for teleoperation. The experiments prove the method to be an accurate and effective means of teleoperation.

KEYWORDS: Teleoperation, Virtual environment, 3D point cloud, Dynamic modelling

1. Introduction

Research on teleoperation has been conducted for many decades. Teleoperation refers to controlling robot interaction within its environment from a remote distance, which helps humans accomplish unreachable tasks. Currently, teleoperation is mainly deployed in outer space robot manipulation, deep ocean exploration, remote surgery, etc. The DARPA Robotics Challenge (DRC) focusses on controlling a humanoid robot to complete challenging tasks that emulate disaster response scenarios, such as opening a door or inserting a plug, which are meaningful for rescuing humans in natural disasters.[1] Therefore, it is important to propose an effective teleoperation strategy. A typical teleoperation system is mainly composed of five components: the human operator, the master system, the communication line, the slave robot system and the environment, as shown in Fig. 1. The human operator interacts with the master system at the local side mainly by using a haptic device, such as PHANToM Desktop or the Novint Falcon. The command, mainly motion and force, is transmitted to the slave robot system through the communication line. On the remote side, the slave robot system carries out the command accordingly. The force and visual information captured by multiple sensors during the interaction with the environment is fed back to the local side. In this way, the human operator can perceive the real interaction with the environment through sensory feedback.

However, in an actual teleoperation system, the substantial communication delay distorts the presence of the human operator and deteriorates the transparency and stability of the teleoperation system. To eliminate the negative influences of communication delay, diverse control techniques have been applied to teleoperation. An overview of early research is described in ref. [2]. Some recent surveys on teleoperation can be found in refs. [3, 4]. This previous work is mainly based on passivity and scatter theory, maintaining passivity under communication delay and wave variable encoding of the force and motion information. When the communication delay is tens to hundreds of milliseconds,[5]

* Corresponding author. E-mail: a.g.song@seu.edu.cn

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

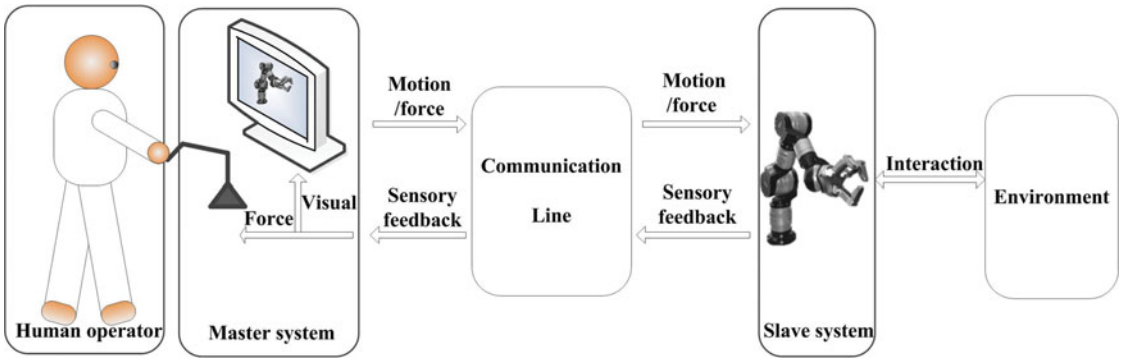


Fig. 1. Typical structure of a teleoperation system.

the system can remain somewhat stable and transparent. However, as the communication delay increases, the performance and presence are somewhat poor in the teleoperation.

A predictive display approach has been proposed[6] to improve the transparency during the task performance. The early idea is to overlay a simulated graphic model on the delayed video images[7,8] from the remote side so that operators are able to pre-operate the virtual robot in the predictive environment transparently and directly. This method enables operators to locally view the virtual robot motion relative to the robot in the video images and avoid some risky operations. An extension of the predictive display method has been proposed in ref. [9]. A 3D geometric environment model is built by using a stereo camera, which is severely time-consuming. Although this approach could compensate for the transparency of vision, it is difficult to provide operators with force feedback from the interaction with the predictive environment.

Recently, teleoperation with force feedback based on a virtual environment (VR) has been proposed to tackle both long time delay and poor virtual transparency.[10] A VR model is created on the local side computer according to the sensory information. Virtual interaction and force reflection are provided to the human operator through the virtual reality environment. The sensory information is mainly composed of visual data and force data, which are used to calculate the geometric and physical parameters of the real environment. Nevertheless, teleoperation is mainly performed by the human operator for the operations in an unknown or partially known environment. Moreover, the environment is usually unstructured. It is considerably tough to acquire the precise geometric and physical parameters of the real environment through sensory data. Therefore, an accurate VR exactly consistent with the real environment cannot always be guaranteed. To overcome this challenge, researchers try to develop an intelligent controller[11] to avoid the model errors between the VR and the real environment. However, it is critical to build a VR that closely resembles the real one.

Recent years have witnessed the development of model-mediated teleoperation,[12] in which the environment is continuously updated as the slave robot interacts with the environment. This method is based on point clouds to apply the estimated model parameters in the VR[13] to realize teleoperation. However, the geometric property is mainly based on a local point cloud, which lacks the detailed information of the full view of the real environment. Because the teleoperation system is human-in-loop, it is quite important to design an efficient human–robot interaction method. Types of human–robot interaction methods from the DRC Trials were analysed.[14] However, the research in the trials is based on the pre-known layouts. In contrast to the conventional way to build the VR, a 3D-point-cloud augmented virtual environment (PCAVR) teleoperation method is proposed in this paper, which is specifically designed for the interaction task in earth orbit teleoperation, such as contacting with objects, moving objects, etc. First, a 3D full view point cloud (FVPC) of the real environment is built by registering multi-point clouds from multi-view around the manipulation platform. Second, a virtual robot on the master side is registered in the 3D FVPC transmitted from the slave side. Third, haptic rendering is applied to the FVPC through the dynamic modelling based on the online estimated parameters. The human operator can then control the virtual robot to directly interact with the FVPC. There are several advantages to use this approach. First, human operators are able to directly manipulate the virtual robot in the PCAVR environment, which can provide a view of the real environment from multiple perspectives, compared with the method of overlaying a geometric model

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

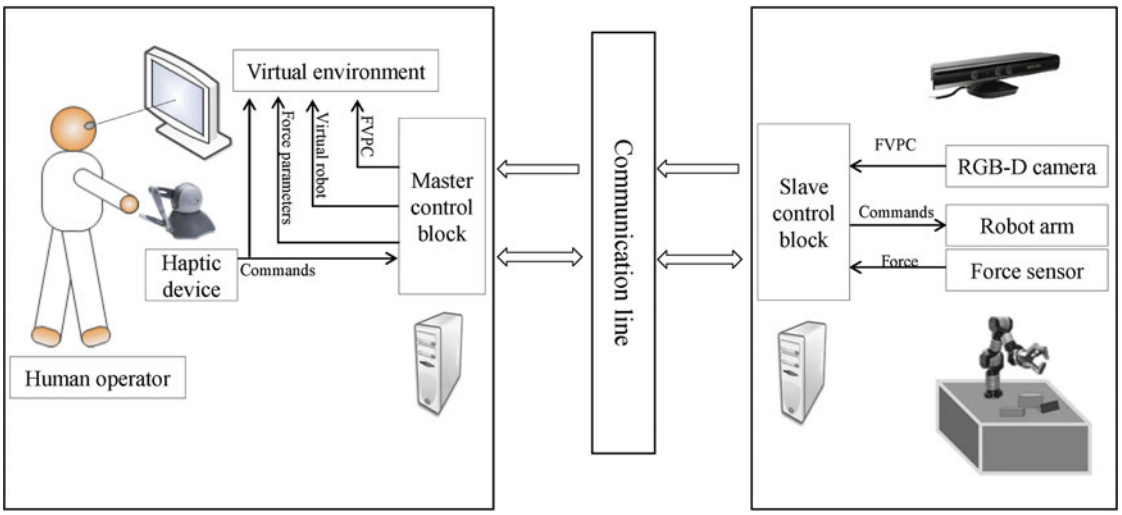


Fig. 2. Overview of the PCAVR teleoperation system.

on video images. Second, without computing geometric information such as the object position, object dimensions, and so on, operators can interact with FVPC using the haptic device if the FVPC is precisely consistent with the real world. Third, the haptic rendering of FVPC based on real-world dynamic modelling can enhance the transparency of the teleoperation.

Our main contributions of the work are as follows:

(1) To propose a method for building an FVPC VR based on 3D-point-cloud registration.
(2) To use haptic rendering on a point cloud for simulating the real world based on the weighted sliding-average least square (wSALS) parameter estimation algorithm.

The remainder of the paper is organized as follows: Section 2 introduces the overall design of the PCAVR teleoperation system. Section 3 describes the FVPC building method, which includes the FVPC registration method and the haptic model rendering approach in detail. Section 4 shows the experiments and the results. Finally, some conclusions and future work are drawn in Section 5.

2. PCAVR Teleoperation System

Figure 2 depicts the overview of the PCAVR teleoperation system. The human operator interacts directly with the VR, which represents the remote environment, through the haptic device in this system, using the PHANTOM Omni device. The haptic device, the six degree-of-freedom positional sensing, and three degree-of-freedom force-output are used for controlling the virtual robot arm and acquiring the haptic reflection from the interactions in the VR. In parallel, commands are transmitted to the slave control block to control the real robot arm by the communication line.

To build the PCAVR teleoperation system, the RGB-D cameras mounted on the robot arm are applied to acquire multi-perspective point clouds. The point clouds contain the RGB information and coordinate position data relative to the camera of each frame. They present a full 3D view, which is consistent with the real environment in geometry. The point-cloud registration technique is used to obtain an FVPC model for the virtual scene of the whole real environment. This technique is critical and plays a fundamental role in building a perfect FVPC model for teleoperation. When the FVPC is calculated on the remote side, the FVPC data are shared with the master control block. The virtual robot arm is superimposed in the FVPC model to build a PCAVR teleoperation display by the coordination transformation. In this way, the virtual robot provides the human operator with a full perception of its spatial location and flexible viewpoint options. The PCAVR teleoperation display can provide the human operator with detailed 3D observation of the interaction between the virtual robot and the FVPC model of the real environment.

With a full view of the FVPC, the interaction with force presence plays an essential role in building the telepresence of the human operator. In this system, a mass–damper–spring model is applied to

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

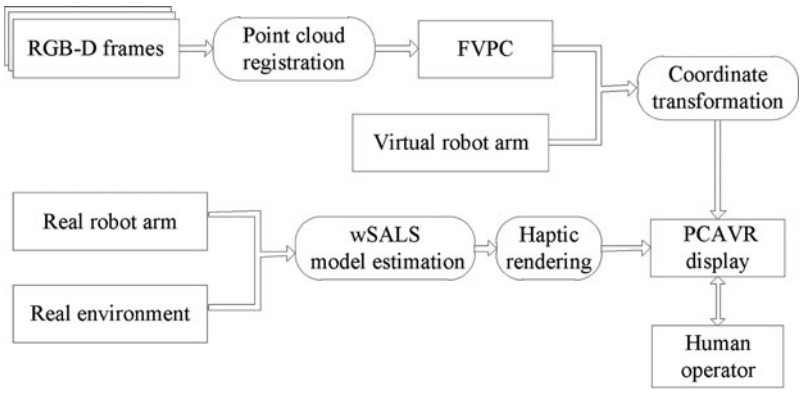


Fig. 3. Procedure of the PCAVR display construction method.

simulate the real environment. The model parameters estimated from the real environment are used to achieve haptic rendering on the FVPC. During the interaction with the real environment, an online dynamic properties estimation method based on the wSALS algorithm is used to calculate the model parameters on the remote side, and the estimated parameters are transmitted to the master side to update the VR. Therefore, the whole PCAVR-based teleoperation system is built in three steps: FVPC model construction based on the incremental registration trajectory of the point cloud, wSALS-based environment modelling and local haptic rendering on the point cloud.

3. PCAVR Display Construction Method

Recent years have witnessed the development of 3D vision in the popular areas of robotics, medical measurement, etc. 3D dense depth data are easily acquired depending on the sensing devices, such as laser measurement systems (LMS), radar and time-of-flight cameras (TOF). These sensors are applied in many teleoperation systems.[13,15,16] However, in these systems, the sensors are used only for acquiring the local information for manipulation. Kaul[17] proposed a laser-based mapping solution for the MAV platform that produced accurate and detailed 3D-point-cloud maps. However, it is unsuitable for robot teleoperation because it does not consider the size of point cloud data in the communication line. In ref. [1], an operator control station (OCS) is developed to realize important object modelling by using 3D templates, which is not applicable for an unknown environment. In our system, Kinect sensors are applied on the slave side to capture the real environment. Kinect sensors are chosen because Kinect consists of an RGB camera and a 3D depth camera. Nevertheless, a single frame of the sensors is insufficient to acquire the whole scene of the real environment. Therefore, a point-cloud registration method based on RGB-D data is proposed to acquire an FVPC of the real environment by the continuous registration of the RGB-D frames.

The whole procedure of the PCAVR display construction method is described in Fig. 3. RGB-D frames captured by Kinect sensors are registered into one consistent global model. This global model is consistent with the coordinates of the Kinect. By coordinate transformation, the global model is transferred into the real robot arm world model to combine the FVPC and the virtual robot arm in one VR for PCAVR display. Later, the human operator operates the virtual robot arm through the haptic device, and the commands are also transmitted to the slave side. The force modelling parameters during interaction are acquired by the wSALS algorithm, which is applied to the haptic rendering in the PCAVR environment.

3.1. FVPC model construction based on the incremental registration trajectory of the point cloud

Because the Kinect sensors are applied in this system, there are two issues of the FVPC model construction by Kinect sensors, as shown in Fig. 4: (1) The raw data of the point cloud are rather noisy, and there are some data absent for a whole scene because of the erroneous structure light reflection and the invalid measurement. (2) One frame captures only partial views of one target object. An effective and accurate point-cloud registration trajectory plays a fundamental role in constructing a PCAVR display.

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

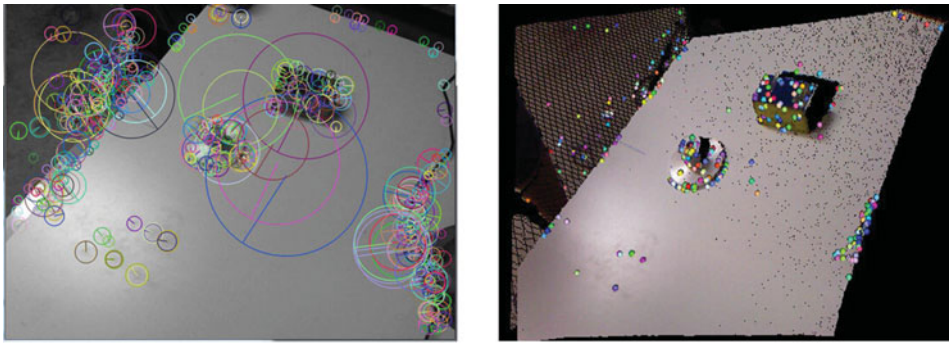


Fig. 4. (a) There are many noises and absences in the raw data; (b) Only a partial view of a target object is obtained.

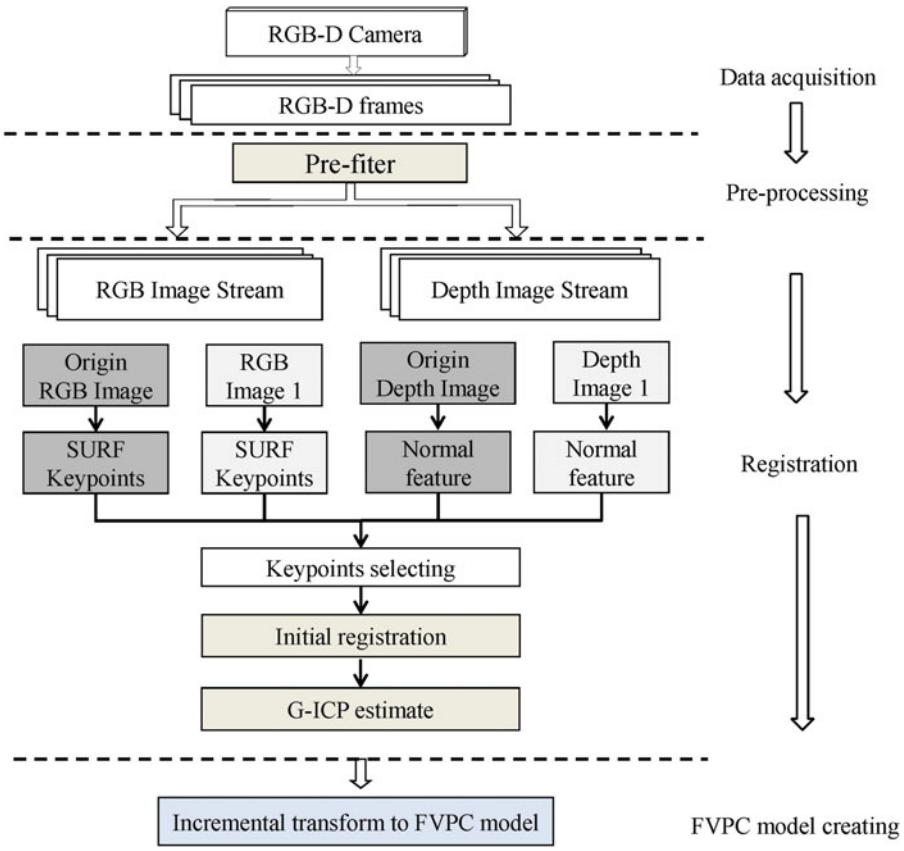


Fig. 5. Detailed procedure of the incremental point-cloud registration trajectory.

The challenges mentioned above lead to some difficulties in constructing an accurate FVPC model for teleoperation. In this system, a novel incremental point-cloud registration method is particularly proposed for FVPC model construction. Figure 5 presents the detailed procedure of the incremental point-cloud registration trajectory.

Point-cloud data are typically noisy, and measurement errors may lead to a poor effect on later registration. It is essential to reduce the noise effect of the point-cloud data. To realize a real-time pre-filter, a median filter, which is a typical non-linear filter method,[18] is applied to remove the outlier noise in point-cloud data. Here, a 5×5 median filter is used.

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

Research on point-cloud registration methods has been performed for several decades. The goal of this research is to find a Euclidean transformation between a set of point clouds from different views that contains a rotation and translation (R, t).[19] In Eq. (1), R is a 3×3 matrix representing the rotation, and t is a 1×3 column vector representing the translation. We will find the (R, t) values that minimize the error, assuming two point clouds P and Q, $p_n \in P$, and $q_n \in Q$.

$$E(R, t) = \min_{R,t} \sum_{n} \|q_n - (R p_n + t)\|^2. \tag{1}$$

The common technique used is the iterative closest point (ICP) algorithm.[20] This method is suitable for a cloud with a good initial alignment. Although the Kinect sensors work at 30 *FPS* and the neighbour point cloud may change little, it is not necessary to register each point cloud to the FVPC, which would cost much routine time in the system. Therefore, point clouds are captured when the real robot arm moves a constant distance. Because the neighbour point cloud may not have a good initial alignment, in the proposed system, we apply three steps in calculating the transformation vectors.

First, detect the keypoints in the point clouds. Here, we first use the SURF (speeded up robust features) keypoint detection method to find the RGB keypoints in the RGB images. SURF is fast and scale-invariant, which is practical in the incremental registration.[21] Although SURF keypoints in RGB images are detected, these points are not always available in the relative depth image. We then combine the normal feature of the point cloud to make a detailed selection for the final keypoints chosen.

Second, apply the Random Sample Consensus (*RANSAC*) to obtain an initial registration based on the final keypoints.

Third, apply the G-ICP algorithm on the whole point cloud. By iteratively proceeding to minimize the error between a pair of point clouds, the transformation for the point clouds is determined.

When the transformation between the latest point cloud and the current point-cloud model is acquired, the point-cloud model is incrementally completed.

3.1.1. Keypoint selection. The point-cloud frames are composed of RGB image data $I\ (u, v)$ and depth image data $D\ (u, v)$. The methods for detecting keypoints depend on the data used. In the proposed system, we first detect the keypoints in the RGB image data $I\ (u, v)$. The SURF detector is scale- and rotation-invariant and is computed much faster than other methods. Because the OpenCV library provides a codec implementation of the SURF detector, Visual C++ and OpenCV are used to detect the SURF keypoints in $I\ (u, v)$.

However, the relative depth data of SURF keypoints may be empty. To obtain the keypoints, not only scale- and rotation-invariant in *I (u, v)* but also geometric-invariant in *D (u, v)*, it is essential to consider both the RGB features and depth features. Therefore, the geometric features are extracted for determining the final keypoints. The rules are described as follows.

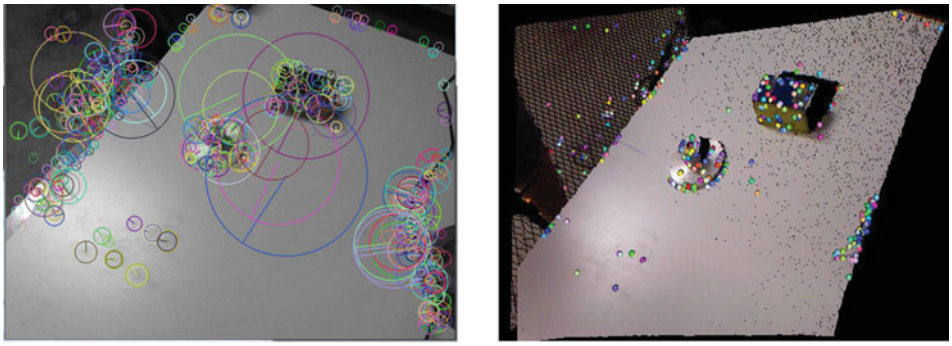
Rule 1, discard the keypoints without depth data. If the relative depth data are empty, the keypoint will not be included in the final keypoints.

Rule 2, keep the keypoints whose angles of deviation with the r-neighbours are large enough. For this rule, three steps are carried out. First, calculate the normal of the remaining keypoints. Here, define a keypoint D, and the neighbours are limited by the radius r. The solution for the normal vector $\boldsymbol{n} = \{n_0, n_1, n_2\}$ is found by analysing the eigenvalues and eigenvectors of the covariance matrix C, as in formula (2).[22]

$$C = \frac{1}{r} \sum_{i=1}^{r} \left(d_i - \bar{d}\right) \bullet \left(d_i - \bar{d}\right)^T, C \bullet \vec{V}_j = \lambda_j \bullet \vec{V}_j,\ j \in \{0, 1, 2\}. \tag{2}$$

If $\lambda_{min} > 0$, the corresponding eigenvector could be set as $\boldsymbol{n} = \{n_0, n_1, n_2\}$. Second, obtain the normal vector of the neighbour points around the keypoints, and obtain the angle of deviation from the keypoint normal. In formula (3), ω is defined as the deviation angle, the vector $\boldsymbol{n} = \{n_0, n_1, n_2\}$

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press



(a) (b)

Fig. 6. (a) The number of valid keypoints is 546; (b) The number of valid keypoints is 94.

as the neighbour point normal and $\boldsymbol{n}_k = \{n_0, n_1, n_2\}$ as the keypoint normal.

$$\omega = \arcsin\left(\sum_{i=0}^{2} n_i n_{ki} \Big/ \sqrt{\sum_{i=0}^{2} n_i^2} \sqrt{\sum_{i=0}^{2} n_{ki}^2}\right). \tag{3}$$

Third, set an approximate ϵ_k to keep the keypoint with a larger standard deviation δ. In formula (4), μ_k is the average deviation angle of the keypoints, and δ_k is its standard deviation.

$$\delta_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_i - \mu_k)^2}. \tag{4}$$

The keypoints detected in the RGB image data using the SURF detector are shown in Fig. 6(a), whereas the final keypoints reserved by the regulations of the geometric features are shown in Fig. 6(b). The number of final valid keypoints is 94. As the keypoints are selected, the final keypoints are used for the transformation computing.

3.1.2. Incremental registration to an FVPC. A well-known approach to obtain the correct corresponding pair of points is the *RANSAC* algorithm.[23] After obtaining the valuable keypoints, we use a new descriptor that supplements the SURF descriptor with the geometric feature of the parameters μ_k and δ_k.. By matching the descriptors by *RANSAC*, we can obtain the relative correct correspondent matched pairs. With the correspondent pairs, apply *RANSAC* to calculate the transformation vector (*R, T*). The transformed point cloud (L') is obtained as formula (5).

$$L' = R_i \bullet P + T_i. \tag{5}$$

We can obtain a relatively better initial registration, whereas a fine registration is still essential for an accurate registration for the partly overlapped point-cloud data. G-ICP, proposed in ref. [24], combines "the ICP" method with the "point-to-plane ICP" algorithm into a single probabilistic framework. This algorithm is popular in practice with high accuracy.

After the fine registration, the transformed L' is overlaid with *F*. Because the former point cloud stands for an FVPC, it is iteratively registered by the later point cloud and overlaid to form a new FVPC. However, the new FVPC has plenty of redundancy points. Therefore, an adaptive down-sampling algorithm is applied to the FVPC.

The overlaid FVPC is defined as P. The detailed algorithm is shown below. The constraint condition depends on the number of neighbours. If the number of points is large enough, we apply a

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

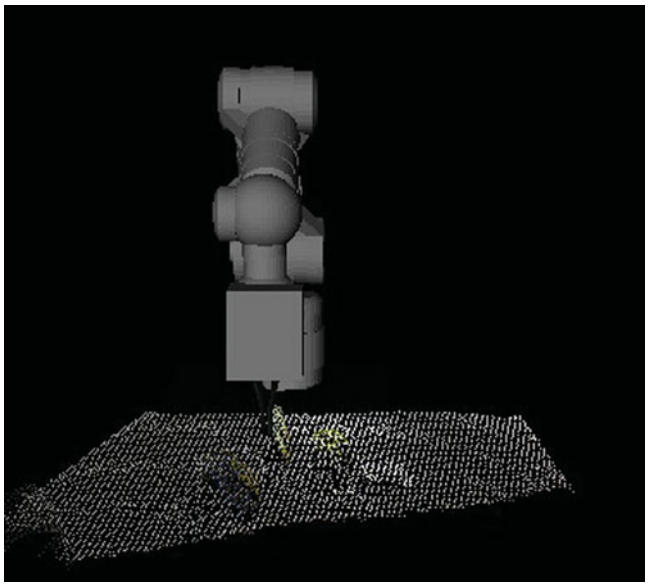


Fig. 7. Scene of the point cloud with the robot model fusion.

voxel grid filter on this region. When all points are processed, the filtered point cloud *PN* is created after the adaptive sampling. *PN* is determined as a new FVPC.

Algorithm Adaptive sampling (FVPC P, FVPC PN)
1:**for** each point P^i in P **do**
2: N^i = get_neighbors_in_radius (P^i, P, R)
3: Q^i = count (N^i)
4: **end for**
5: N_LIMIT = get_middle_num (Q)
6: **for** each point P^i in P **do**
7: **if** $Q^i > N_LIMIT$ **then**
8: PN^j = voxel_grid_filter (N^i, d)
9: **else if** $Q^i <= N_LIMIT$ **then**
10: $PN^j = Q^i$
11: **end if**
12: **end for**
13: $P = PN$

3.1.3. Model fusion. Although the FVPC is described in the Kinect coordination, it is critical for transforming the FVPC to the real-world coordination. In the PCAVR display, we set the robot arm base as the origin of the world coordination and the robot arm coordination as the world coordination. Assume a point in the FVPC described as the vector $(x, y, z)^T$, and the transformation from the Kinect coordination to the real-world coordination is R and T. Therefore, the point in the FVPC is transformed to the real-world coordination by formula (6).

$$(x_w, y_w, z_w)^T = R \bullet (x, y, z)^T + T. \tag{6}$$

The scene of the point cloud with the robot model fusion is shown in Fig. 7.

3.2. Haptic rendering for FVPC model

Haptic interaction with the point cloud has been proposed by Fredrik[25] and extends the existing method for point-based haptic rendering based on a proxy. Constraint-based haptic rendering on point clouds is described in ref. [26], and it is applied for teleoperated robot grasping.[27] However, the physical properties of interaction often vary in a real teleoperation system because the manipulation environment is non-static. It is obligatory to apply the nearly true dynamic model of real-world modelling for haptic rendering. Owing to the development of a multi-dimensional force sensor for

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

the robot,[28] the contact force between the robot and the object can be more accurately measured than before. It is essential to propose an effective algorithm to simulate the contact process depending on the force data and geometric data. In this paper, a wSALS algorithm is used to calculate the dynamic parameters of the environment.

3.2.1. Force data pre-filter. The noises and disturbances of force sensors are unavoidable, which could lead to a poor effect on the later modelling. It is critical to reduce the noises and disturbances of force data. Because the data from the force sensors used in this system (mentioned later) are relative steady after the load is stable, the disturbance of the temperature does not cause a bad effect. For the noises caused by the measurement method, the vibration method, and other reasons, a force data pre-filter mainly based on singular point removal, and the moving average method is applied to obtain smooth and accurate data for later processing.

3.2.2. Weighted SALS algorithm. Mass, damper and spring are three basic properties of a mechanical dynamic system.[29] The behaviour of such a system is generally represented by models composed of series arrangements of masses, dampers and springs, as linear elements. The proposed wSALS algorithm works in real time. For online computation, it is critical to consider the complexity of the algorithm for model parameter estimation. The mass–damper–spring model is a proper way to describe the interaction with compliant objects,[30] and its simplicity has made it a popular and effective choice.[31] Therefore, in the proposed system, a mass–damper–spring model is used to simulate the contact procedure between the robot arm and the objects. Likewise, the VR is accordingly constructed. In the VR, the virtual force is generated by the contact between the haptic interface point (HIP) and the virtual object. The movement of the HIP is inputted by the haptic device, which presents the movement of the virtual robot. The mass–damper–spring model of the contact procedure is usually expressed as formula (7).

$$f_v = m_v\ddot{x} + b_v\dot{x}_v + k_v x_v, \tag{7}$$

where f_v denotes the virtual force between the HIP and the objects, and m_v, b_v and k_v represent the mass, damper and stiffness of the VR, respectively. The parameters are equal to the parameters of the real environment as presented in formula (8).

$$\begin{cases} m_v = m_e \\ b_v = b_e \\ k_v = k_e \end{cases}. \tag{8}$$

In ref. [32], our lab presented a sliding average least square algorithm to identify the dynamic parameters of the slave environment. However, this method cannot treat the sudden change, which causes major error and instability. In this paper, a wSALS approach is proposed to identify the parameters online, which is useful to acquire the haptic parameters of the FVPC.

According to the sliding least square method, suppose N is the number of sampling points estimated, and $\hat{m}_e$, $\hat{b}_e$ and $\hat{k}_e$ are the estimations of m_e, b_e and k_e, respectively. A weighted variable is adopted to balance the contribution to parameter estimation.

$$\begin{bmatrix} x_{ew}(i) \\ f_{ew}(i) \end{bmatrix} = \frac{i}{N}\begin{bmatrix} x_e(i) \\ f_e(i) \end{bmatrix}, \tag{9}$$

$$\begin{bmatrix} \hat{m}_e(i) \\ \hat{b}_e(i) \\ \hat{k}_e(i) \end{bmatrix} = \begin{bmatrix} \sum_{i=t-N+1}^{t} \ddot{x}_{ew}^2(i) & \sum_{i=t-N+1}^{t} \ddot{x}_{ew}(i)\dot{x}_e(i) & \sum_{i=t-N+1}^{t} \ddot{x}_{ew}(i)x_{ew}(i) \\ \sum_{i=t-N+1}^{t} \dot{x}_{ew}(i)\ddot{x}_{ew}(i) & \sum_{i=t-N+1}^{t} \dot{x}_{ew}^2(i) & \sum_{i=t-N+1}^{t} \dot{x}_{ew}(i)x_{ew}(i) \\ \sum_{i=t-N+1}^{t} x_{ew}(i)\ddot{x}_{ew}(i) & \sum_{i=t-N+1}^{t} x_{ew}(i)\dot{x}_{ew}(i) & \sum_{i=t-N+1}^{t} x_{ew}^2(i) \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{i=t-N+1}^{t} \ddot{x}_{ew}(i)f_{ew}(i) \\ \sum_{i=t-N+1}^{t} \dot{x}_{ew}(i)f_{ew}(i) \\ \sum_{i=t-N+1}^{t} x_{ew}(i)f_{ew}(i) \end{bmatrix}, \tag{10}$$

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

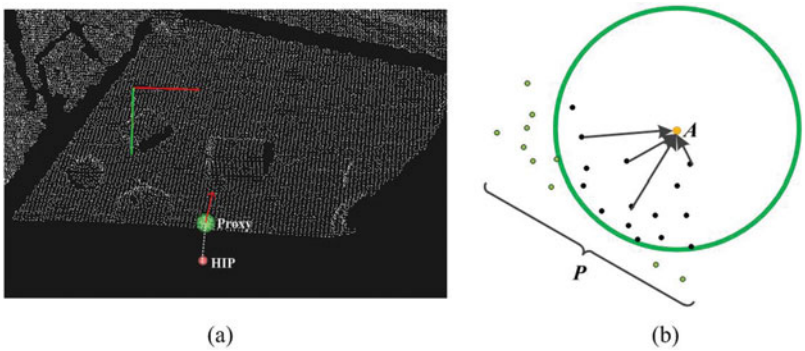


Fig. 8. (a) The proxy-HIP approach; (b) The local surface normal estimation method.

$$\left[\hat{Z}_e\,(t)\right] = [A\,(t)]^{-1}\,[C\,(t)]\,. \tag{11}$$

The matrixes $[A(t)]$ and $[C(t)]$ can be calculated in real time by the sliding method, according to formulas (12) and (13).

$$\begin{aligned} a_{i,j}\,(t+1) &= \sum_{l=t-N+2}^{t+1} x_{ew}^{(3-i)}\,(l)\,x_{ew}^{(3-j)}\,(l) \\ &= a_{i,j}\,(t) + x_{ew}^{(3-i)}\,(t+1)\,x_{ew}^{(3-j)}\,(t+1) - x_{ew}^{(3-i)}\,(t-N+1)\,x_{ew}^{(3-j)}\,(t-N+1) \\ i &= 1,2,3;\; j = 1,2,3, \end{aligned} \tag{12}$$

$$\begin{aligned} c_i\,(t+1) &= \sum_{l=t-N+2}^{t+1} x_{ew}^{(3-i)}\,(l)\,f_{ew}\,(l) \\ &= c_i\,(t) + x_{ew}^{(3-i)}\,(t+1)\,f_{ew}\,(t+1) - x_{ew}^{(3-i)}\,(t-N+1)\,f_{ew}\,(t-N+1) \\ i &= 1,2,3. \end{aligned} \tag{13}$$

The estimated parameters of the real environment are then calculated.

3.2.3. Haptic rendering. Similar to ref. [25], the surface is represented by local points for real-time computing. A proxy-HIP approach is employed to estimate haptic interaction. As shown in Fig. 8(a), the red sphere stands for the HIP, whereas the green one represents the virtual proxy. The radius of the proxy is set as R. The points within R ranges are applied to calculate the surface normal and proxy position. If there are no points within R ranges, the interaction condition is not contacted. When there are points within R ranges, the proxy is in contact with the environment.

When the proxy is in contact with the point cloud, the local surface normal is estimated as follows. Given the point set P within R around the proxy and the centre position A, calculate the surface normal n. As shown in Fig. 8(b), the yellow point is the centre A, and the black points are the set P in the proxy range (the green circle). The surface normal is an average of the vectors of all normal vectors of the K-neighbours, whereas the normal vector of each neighbour point is calculated according to Section 3.1.1. However, vectors of the points that are far from the centre contribute less to the estimation of the surface normal. Therefore, a weight function w_i is used for the contribution weight, which requires less computational cost. The weight function is a linear relation about the distance between the HIP and the proxy. Define r as the distance from the point to the centre point, R as the radius of the K-neighbours and n_{pi} as the normal vector of each

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

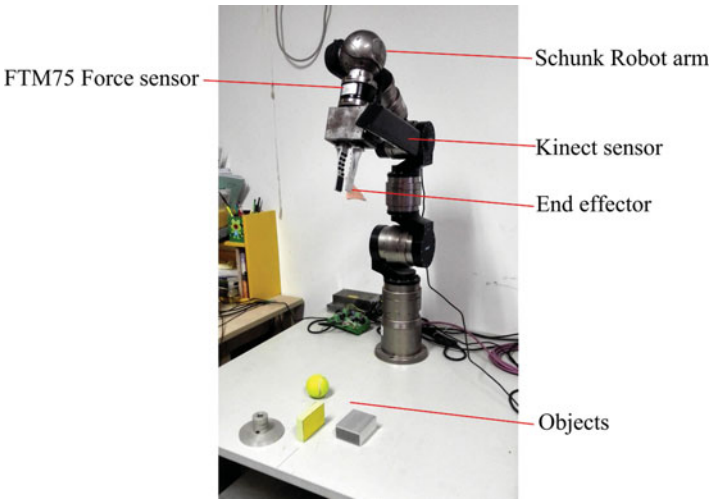


Fig. 9. The system setup.

neighbour point.

$$w = \begin{cases} 1 - \dfrac{r}{R}, r \le R \\ 0, r > R \end{cases}, \tag{14}$$

$$\vec{n} = \frac{\sum_i w_i \overrightarrow{n_{P_i}}}{\left\| \sum_i w_i \overrightarrow{n_{P_i}} \right\|}. \tag{15}$$

The surface normal is calculated by formula (15), and m_v, b_v and k_v are acquired by the SALS algorithm from the slave side in real time. Therefore, the virtual force can be calculated accordingly. The force on the haptic device is rendered at f_d =1 kHz. Define vector d as the vector from the proxy to the HIP; the force rendered on the haptic device is calculated as

$$\vec{F} = \vec{n}\left(m_v f_d^2 \left(\left\|\vec{d}_t\right\| - 2\left\|\vec{d}_{t-1}\right\| + \left\|\vec{d}_{t-2}\right\|\right) + b_v f_d \left(\left\|\vec{d}_t\right\| - \left\|\vec{d}_{t-1}\right\|\right) + k_v \left(\left\|\vec{d}_t\right\|\right)\right). \tag{16}$$

The force F is rendered only when the proxy is in contact with the environment, and the direction from the HIP to the proxy is consistent with the surface normal.

4. Experimental Evaluation

4.1. System setup

The experiment is carried out on the 7-DOF SChunk Robot Arm on the remote side, whereas on the master side, a PHANTOM Omni device that can provide 6-DOF position measurement with force feedback is used to finish the haptic interaction. One Kinect sensor is mounted beside the end joint of the robot arm. The force sensor FTM75 is set at the end of the robot arm. The remote side communicates with the local side by the TCP-IP communication protocol. The software is based on ROS, the PCL library and the OpenCV library on the remote side to finish the FVPC construction, whereas OpenGL is used to render the 3D graphics on the master side. The computers run at 3.2 GHz. The real system setup is shown in Fig. 9.

4.2. FVPC model construction evaluation

The FVPC model plays a fundamental role in the proposed teleoperation system. The accuracy of the registration of the FVPC model construction is the most important factor to be considered. There are still several other factors that affect the construction procedure, such as computation speed

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

Table I. Relative transformation parameters.

Point cloud	Rot X (°)	Rot Y (°)	Rot Z (°)	Trans X (mm)	Trans Y (mm)	Trans Z (mm)
Pair 1	0	5	8	1.4	1.5	2.2
Pair 2	5	8	0	2.5	2.3	3.1
Pair 3	5	8	8	2.1	2.5	2.3
Pair 4	0	12	14	3.2	2.1	3.5
Pair 5	10	12	0	2.1	2.8	3.8
Pair 6	10	0	14	3.5	3.1	2.2
Pair 7	0	15	18	2.2	2.1	0.3
Pair 8	18	15	0	3.6	2.3	4.1
Pair 9	18	0	18	2.7	3.1	4.3
Pair10	10	8	13	3.4	3.5	4.2

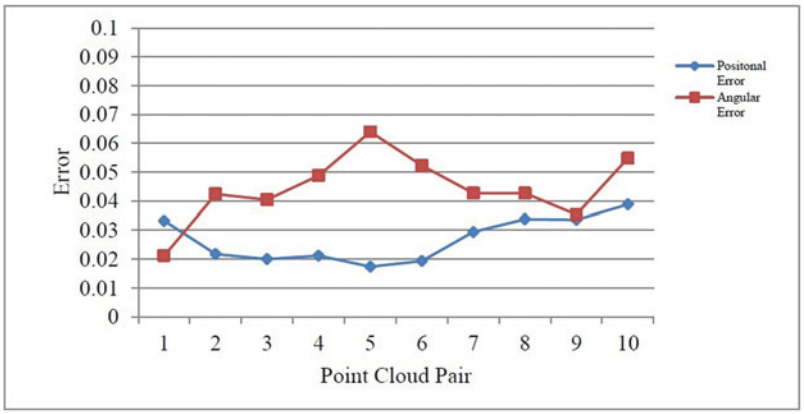


Fig. 10. Rotation and transformation errors.

and registration accuracy. We focus on testing the registration accuracy. The measurements used to measure the accuracy are the rotation error and the transformation error. To evaluate the registration accuracy, we test the algorithm on 10 known transformation point clouds and apply the algorithm during the movement of the robot arm to confirm its effectiveness.

First, 10 pairs of point-cloud frames are acquired from the Kinect sensor during the robot arm movement. Supposing that point cloud A is randomly acquired, the Kinect sensor is transformed to a new position by specific rotation and transformation parameters for controlling the robot arm, and then the point cloud A' is obtained. To test the robustness of the registration method, the transformation of each pair of point clouds was set to be different. The relative transformation parameters are shown in Table I.

The rotation angle and translation distance are randomly chosen in the set range. The rotation and transformation errors are calculated in the registration of each point cloud and the transformed one.

Figure 10 depicts the average error of the positional and angular transformation for 10 point clouds. The error of registration parameters is less than 0.07, and the transformation estimation is more accurate if the movement is more slight. Therefore, this registration algorithm can meet the need for the procedure of FVPC model construction. Because the adaptive sampling algorithm for the FVPC is specifically proposed for the teleoperation application scenario, it is essential to test the effectiveness of this method. Here, we list the valid number of registered point clouds with the adaptive sampling algorithm (marked as AS) compared to that without the algorithm (marked as nonAS). As we can see from Table II, the redundant points in the point clouds are removed accordingly.

In the real system, the robot arm moves around the environment to finish the FVPC model construction. The procedure of point-cloud registration is shown in Fig. 11(a)–(e). Five point clouds shown are captured during the FVPC model construction. As shown in the figures, there are holes in each point cloud, especially around the objects. Furthermore, one point cloud cannot describe the whole manipulated scene for teleoperation. Applying the proposed FVPC model construction

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

Table II. Valid number of point clouds ($\times 10^4$).

Point cloud	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pair 8	Pair 9	Pair 10
AS	32.15	36.14	37.51	38.45	39.01	38.57	42.11	45.32	44.20	46.58
nonAS	55.41	57.81	57.23	56.89	58.23	59.63	57.53	56.85	58.64	56.45
Differ	23.26	21.67	19.72	18.44	19.22	21.06	15.42	11.53	14.44	9.87

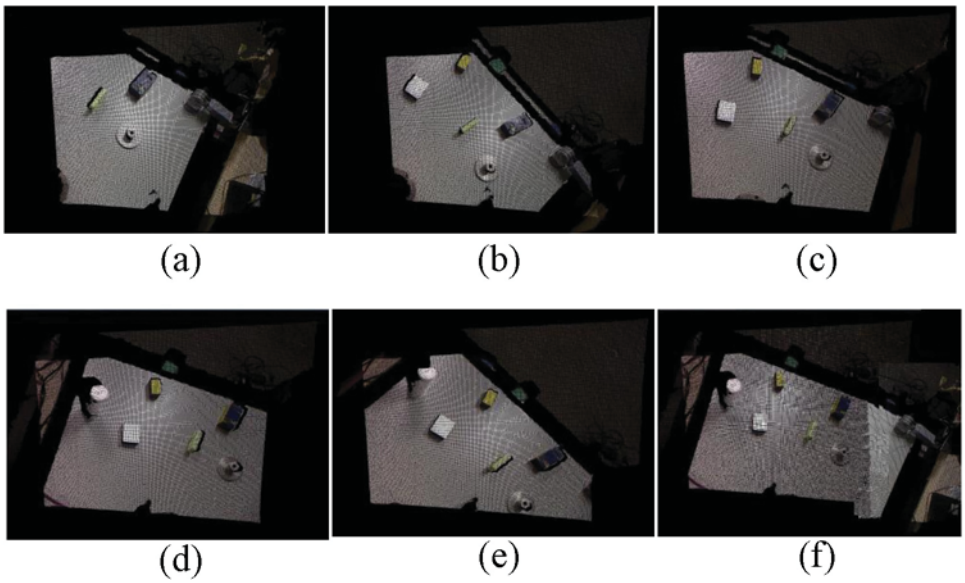


Fig. 11. (a)–(e) Several point clouds; (f) The final registered point cloud.

method, a full view of the manipulated scene is built, and the absent data on the black holes are supplemented. The final FVPC is shown in Fig. 11(f).

4.3. System evaluation

Using the system setup, the system is evaluated by controlling the real robot arm interacting with the real environment through the teleoperation system. First, based on the FVPC model, the whole geometric model of the VR is built. We manipulate the virtual robot arm to interact with the objects in the VR by the haptic device, whereas the real robot arm on the slave side moves according to commands from the VR. The dynamic parameters are estimated online on the remote side. We mainly evaluate the haptic rendering method and the operation efficiency of the teleoperation based on the FVPC augmented VR.

The procedure of the evaluation includes three steps. First, the robot arm is in free space and finishes the FVPC construction process. Second, we control the real robot arm to make contact with object A by directly manipulating the virtual robot arm in the VR. Third, we control the real robot arm to move away from A to contact with another object B. When the contact occurs, the dynamic model parameters are changed and updated. The parameters are transmitted to the VR on the local side. The force feedback of the VR is applied to the haptic device for the operator. The position and force data in three dimensions are recorded for performance assessment. Figure 12 describes the performance in the X, Y and Z directions.

The red line in Fig. 12 shows the data on the slave side, and the blue line indicates the data in the VR. Because the robot arm contacts the objects mainly in the Z direction, we discuss the performance in the Z direction. The virtual robot arm moves first, and the real robot arm follows the trajectory of the virtual robot arm. When the real robot arm contacts object A, the force increases quickly, and the dynamic model parameters are relatively modified. The force feedback in the VR immediately updates. After the real robot arm moves away from object A to object B, the dynamic model parameter abruptly updates because of the sudden contact with object B. The blue line in Fig. 12(f) is the force feedback after the haptic rendering in the FVPC. Figure 12(a), (b) presents the

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

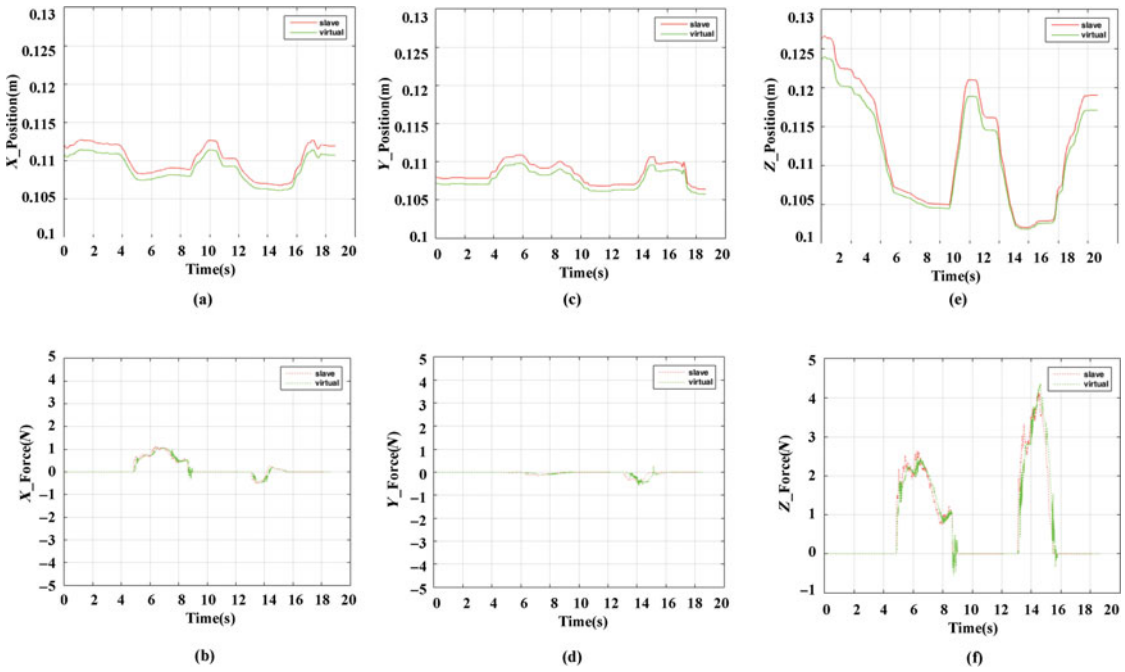


Fig. 12. (a), (b) The interaction performance in the X direction; (c), (d) The interaction performance in the X direction; (e), (f) The interaction performance in the Z direction.

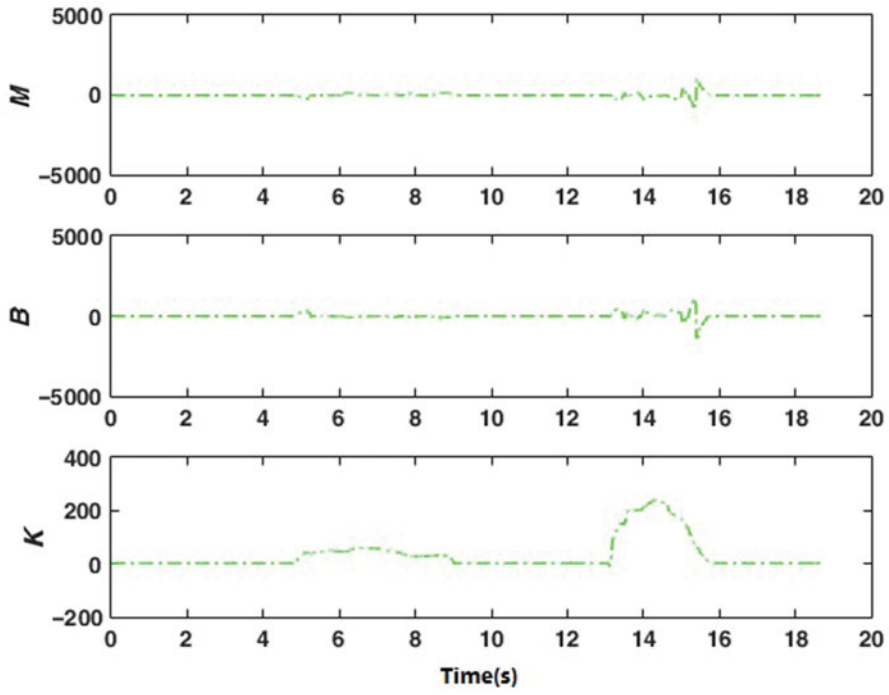


Fig. 13. Online estimated dynamic parameters.

interaction performance in the X direction, and Fig. 12 (c), (d) shows the Y direction performance. The position and force in the two directions slightly vary.

The online estimated dynamic parameters are shown in Fig. 13. The sampling frequency of the force sensor is 2 kHz. The sliding window length is set to 100. Object A is a book pile, whereas object B is a fixed tennis ball. In Fig. 13, M, B and K stand for the mass, damper and spring parameters, respectively. The objects are fixed on the manipulated platform. Parameters M and B

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

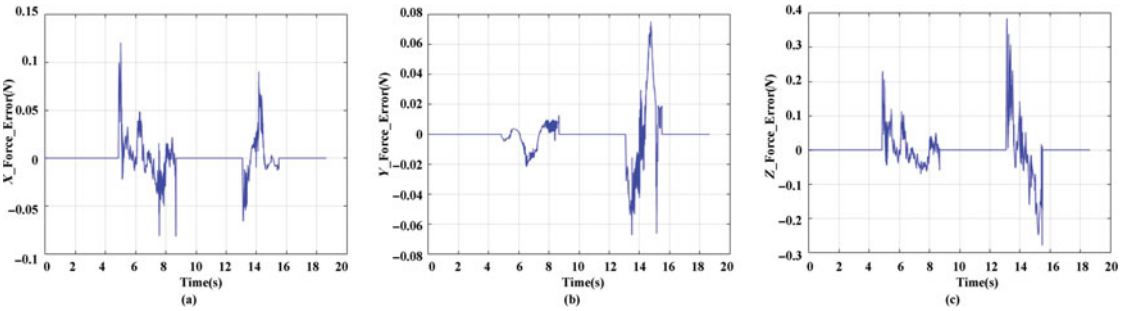


Fig. 14. (a), (b), (c) Force error in the X, Y and Z directions.

change slightly and are almost zero. Parameter K changes mainly twice when the robot arm contacts the objects. Although there is relatively steady contact, the dynamic parameters slightly change. When the slave robot moves in the free space, the contact force is almost zero, and there is no contact with a real object. This could be because the slave robot is making contact with the air in the real environment. Therefore, the parameters M, B and K can be as small as zero when the slave interacts with the environment. Furthermore, because the objects are fixed and the mass of their local surface is relatively small, the parameters M and K are somewhat small values.

Figure 14 depicts the force error in the X, Y and Z directions between the haptic rendering force and the real force. As shown in Fig. 14, the virtual force is almost as precise as the real force. The relative errors are 8.52%, 9.11% and 9.2% in the X, Y and Z directions, respectively. The data assessment demonstrates that the haptic rendering of FVPC based on wSALS is effective.

4.4. Discussion

Using the proposed FVPC model construction method, a VR with geometric parameters is directly built. The registration method is accurately evaluated. The average errors of positional and angular transformation are both low. As shown in the experiments, the FVPC model is constructed during the robot movement around the real environment. In addition, the lost data caused by the Kinect physical principle shown as black holes are compensated by the point-cloud registration. However, there are still a few noises in the FVPC model that may affect the precise manipulation in the future. The physical properties of the environment are estimated by the online wSALS method, which contributes to the haptic rendering in the VR. The precise haptic rendering force can provide the operators with the telepresence sense for teleoperation. However, as shown in Fig. 14, major errors mainly appear when the real robot arm contacts the objects or leaves the objects. Furthermore, the window size of SALS and the force sensor sampling frequency affect the precision of the haptic rendering.

5. Conclusions

In this paper, a novel way to build a VR for teleoperation is described. A 3D-point-cloud augmented VR building method is proposed by registering multi-point clouds from the multi-perspective view around the manipulation platform. The keypoints of the point clouds are detected. The FVPC is built by the proposed incremental registration method. A model fusion method is applied to obtain a whole PCAVR display and to create the geometric modelling of the real environment for teleoperation. The method is proved to be accurate and effective by the experiments. The real-world dynamic parameters are estimated by the weighted SALS method. The algorithm can calculate the dynamic character in real time. A haptic rendering method is applied to the VR to simulate the contact procedure between the real robot arm and the objects. The 3D-point-cloud registration method is proved to be accurate by the experiments. The virtual contact is consistent with the contact in the real world depending on the precise dynamic parameter estimation method and the effective haptic rendering algorithm.

Because teleoperation is mainly applicable for the unreachable tasks in special environments, the robustness of the method is critical. The method should be tested in different types of scenarios. In the future, we will enhance the robustness of the proposed method. Second, as discussed in the experimental section, the adaptability of the online wSALS method and the modelling method will be improved for dynamic simulations.

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

Acknowledgements
This paper is supported by the Natural Science Foundation of China under grant numbers 61325018 and 61403080; the 863 High-Tech Plan under grant number 2013AA013703 and the National Natural Science Foundation of Jiangsu Province under grant number BK20140641.

References

1. S. Kohlbrecher, A. Romay, A. Stumpf, A. Gupta, O. von Stryk, F. Bacim, D. A. Bowman, A. Goins, R. Balasubramanian and D. C. Conner, "Human-robot teaming for rescue missions: Team ViGIR's approach to the 2013 DARPA robotics challenge trials," *J. Field Robot.* **32**(3), 352–377 (2015).
2. T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Trans. Robot. Autom.* **9**(5), 592–606 (1993).
3. L. Chan, F. Naghdy and D. Stirling, "Application of adaptive controllers in teleoperation systems: A survey," *IEEE Trans. Human-Mach. Syst.* **44**(3), 337–352 (2014).
4. C. Passenberg, A. Peer and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems," *Mechatronics* **20**(7), 787–801 (2010).
5. P. Arcara and C. Melchiorri, "Control schemes for teleoperation with time delay: A comparative study," *Robot. Auton. Syst.* **38**(1), 49–64 (2002).
6. A. K. Bejczy, W. S. Kim and S. C. Venema, "The Phantom Robot: Predictive Displays for Teleoperation with Time Delay," *Proceedings of IEEE Int. Conf. on Robotics and Automation*, Cincinnati, America (May 13–18, 1990) pp. 546–551.
7. W. S. Kim and A. K. Bejczy, "Graphics Displays for Operator Aid in Telemanipulation," **In**: *Title of Decision Aiding for Complex Systems*, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Charlottesville, America (Oct. 13–16 , 1991) pp. 1059–1067.
8. J. H. Park and T. B. Sheridan, "Supervisory Teleoperation Control using Computer Graphics," *Proceedings of IEEE International Conference on Robotics and Automation*, Sacramento, America (Apr. 9–11, 1991) pp. 493–498.
9. M. Barth, T. Burkert, C. Eberst, N. O. Stoffler and G. Farber, "Photo-Realistic Scene Prediction of Partially Unknown Environments for the Compensation of Time Delays in Telepresence Applications," *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, America (Apr. 24–28, 2000) pp. 3132–3137.
10. K. Kawabata, T. Sekine, T. Suzuki, T. Fujii and H. A. I. Endo, "Mobile robot teleoperation system utilizing a virtual world," *Adv. Robot.* **15**(1), 1–16 (2012).
11. Z. Jiang, H. Liu, J. Wang and J. Huang, "Virtual reality-based teleoperation with robustness against modelling errors," *Chinese J. Aeronaut.* **22**(3), 325–333 (2009).
12. P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *Int. J. Robot. Res.* **27**(2), 253–262 (2008).
13. X. Xu, B. Cizmeci, A. Al-Nuaimi and E. Steinbach, "Point cloud-based model-mediated teleoperation with dynamic and perception-based model updating," *IEEE Trans. Instrum. Meas.* **63**(11), 2558–2569 (2014).
14. H. A. Yanco, A. Norton, W. Ober, D. Shane, A. Skinner and J. Vice, "Analysis of human robot interaction at the DARPA robotics challenge trials," *J. Field Robot.* 32(3), 420–444 (2015).
15. M. S. Kadavasal, "Sensor augmented virtual reality based teleoperation using mixed autonomy," *J. Comput. Inf. Sci. Eng.* **9**(1), 185–194 (2009).
16. T. Ni, H. Zhang, P. Xu and H. Yamada, "Vision-based virtual force guidance for tele-robotic system," *Comput. Electr. Eng.* **39**(7), 2135–2144 (2013).
17. L. Kaul, R. Zlot and M. Bosse, "Continuous-time three-dimensional mapping for micro aerial vehicles with a passively actuated rotating laser scanner," *J. Field Robot.* **33**(1), 103–132 (2016).
18. J. W. Tukey, *Exploratory Data Analysis* (Addison-Wesley: Cambridge, England, 1977) pp. 210–214.
19. J. Salvi, C. Matabosch, D. Fofi and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.* **25**(5), 578–596 (2007).
20. P. J. Besl and H. D. Mckay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 239–256 (1992).
21. H. Bay, A. Ess, T. Tuytelaars and L. V. Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understanding* **110**(3), 346–359 (2008).
22. R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha and M. Beetz, "Towards 3d point cloud based object maps for household environments," *Robot. Auton. Syst.* **56**(11), 927–941 (2008).
23. C. S. Chen, Y. P. Hung and J. B. Cheng, "RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images," *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(11), 1229–1234 (1999).
24. A. Segal, D. Hähnel and S. Thrun, "Generalized-ICP," *Robot. Sci. Syst.* **2**(4), (2009).
25. F. Ryden and H. J. Chizeck, "A proxy method for real-time 3-dof haptic rendering of streaming point cloud data," *IEEE Trans. Haptics* **6**(3), 257–267 (2013).
26. A. Leeper, S. Chan and K. Salisbury, "Point Clouds can be Represented as Implicit Surfaces for Constraint-Based Haptic Rendering," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, America (May, 14–18 2012) pp. 5000–5005.

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press

27. A. Leeper, S. Chan, K. Hsiao, M. Ciocarlie and K. Salisbury, "Constraint-Based Haptic Rendering of Point Data for Teleoperated Robot Grasping," *Proceedings of the IEEE Haptics Symposium*, Vancouver, Canada (Mar. 4–7, 2012) pp. 377–383.
28. A. Song, J. Wu, G. Qin and W. Huang, "A novel self-decoupled four degree-of-freedom wrist force/torque sensor," *Measurement* **40**(9), 883–891 (2007).
29. E. Doebelin, *System Dynamics: Modelling, Analysis, Simulation, Design*. (CRC Press, United States, 1998).
30. A. Song, D. Morris and J. E. Colgate, "Haptic Telemanipulation of Soft Environment without Direct Force Feedback," *Proceedings of the IEEE International Conference on Information Acquisition*, (Jun. 27–Jul. 3 2005) pp. 21–25.
31. G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mech. Mach. Theory* **37**(10), 1213–1239 (2002).
32. H. Li and A. Song, "Virtual-environment modelling and correction for force-reflecting teleoperation with time delay," *IEEE Trans. Ind. Electron.* **54**(2), 1227–1233 (2007).

https://doi.org/10.1017/S0263574716000631 Published online by Cambridge University Press