2019 IEEE 4th International Conference on Advanced
Robotics and Mechatronics (ICARM)

Robot Teleoperation System Based on Mixed Reality

Congyuan Liang[1], Chao Liu[2], Xiaofeng Liu[3], Long Cheng[4], Chenguang Yang[5,*]

***Abstract*— This work develops a novel robot teleoperation system based on mixed reality. Combined with Leap Motion and HoloLens, the system is able to offer a better operate experience by allowing the operator to teleoperate a robot within a mixed reality scene. Besides, we also design a simple writing task in this paper, which verifies the proposed system's validity.**

I. INTRODUCTION

With the development of the robot manipulators, robots are commonly applied in complex and hash environments, such as space, earthquake rescue, etc. In some particular tasks, it is necessary to control the robot with the help of teleoperation technology[1][2]. Teleoperation is a process that the operator controls the slave robot at a distance. The "distance" here can be understood as a physical distance, the slave robot can be controled remotely, or it can also be understood as a change in scale, for example, a surgery robot may work on a microscopic level[3]. Most of teleoperation system are included with: the human operator, the slave robot, master device, communications and the environment[4]. In these system, the human operator controls the slave robot through the master device. The master device is used to collect the information of the operator's manipulation and then generate commands and transmit it through communications. After that, the slave robot will complete the task of interaction with the environment following the commands.

With the help of Virtual Reality(VR) technology, it is possible to enable a real-time feedback by generating a virtual reality scene. Junshen et.al.[5], have proposed a method to enable the operator to control the pose of a camera by using the Oculus Rift headsets. Salvatore Livatino et.al.[6], have presented a novel Augment Reality(AR) interface. In [7], Yunsick Sung et al. developed a cleaning robot based on virtual reality environment. A teleoperation system based on Mixed Reality(MR) is presented in this article. Similar as Augment Reality(AR), Mixed Reality(MR) is a variation of Virtual Reality(VR)[8][9]. VR technologies are able to immerse the user inside a synthetic environment completely. During the processing, the user can not see the real world around him[10]. Different from VR, AR/MR enable the user to see the real world, with virtual objects In combination with the real world environment, which means that, the real objects in the real world scene and the virtual objects generated are coexisted in the same space. Besides, MR is usually particular refer to the technology achieved with HoloLens, a Head-mounted display(HMD) developed by Microsoft.

Fig. 1. HoloLens[11]

In this paper, Leap Motion, as shown in Fig.2, is used to capture the human operator's hand motion and then generate position information for controlling the slave robot. Developed by Magic Leap Company, Leap Motion is able to trace the user's hands' trail by using infrared LEDs and depth cameras[12].

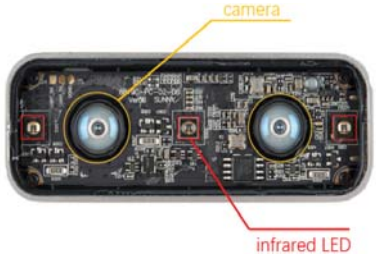


Fig. 2. Leap Motion[13]

Teleoperation belongs to Human-Robot interaction(HRI) [14], which has become a important topic. Many works have been done on this topic, some of them are based on a hand motion tracking device named Leap Motion. In [15], the authors used Leap Motion to trace one of the human operator's

This work was supported in part by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001913, CNRS-NSFC Grants PRC2104, the National Natural Science Foundation of China under Grant 61873268, and in part by the Beijing Municipal Natural Science Foundation under Grant L182060.

[1]C. Liang is with the Key Laboratory of Autonomous Systems and Networked Control, School of Automation Science and Engineering, South China University of Technology, Guangzhou, 510640, China.

[2]C. Liu is with Department of Robotics, LIRMM, UMR5506, University of Montpellier-CNRS, 161 rue Ada, 34095 Montpellier, France.

[3]X. Liu is with Changzhou Key Laboratory of Robotics and Intelligent Technology, Changzhou 213022, China; Jiangsu Key Laboratory of Special Robots, Hohai University, Changzhou 213022, China; College of IoT Engineering, Hohai University, Changzhou 213022, China.

[4]L. Cheng is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.

[5]C. Yang is with the Bristol Robotics Laboratory, University of the West of England, Bristol, BS16 1QY. U.K.

[*]Corresponding author. Email: cyang@ieee.org.

978-1-7281-0064-7/19/$31.00 ©2019 IEEE

finger to imitate the writing motion with a robot. D.Bassily et al.[16] developed a human-machine communication interface between Leap Motion and a 6-DOFs robotic arm. Lorenzo et al.[17] proposed a ROS integrated interface for remote control a robot through the operator's hands motion.

In this work, we employ HoloLens and Leap Motion to develop a novel Human-Robot interface, to enhance the experience of teleoperation.

II. SYSTEM DESCRIPTION

The robot teleoperation system proposed in this paper is shown in Fig.3. The system is consisted with 4 parts: a human operator, a Leap Motion controller, a slave robotic arm and a Microsoft HoloLens. With the help of HoloLens, the human operator is able to observe the slave robot's movement through a real-time mixed reality scene. Synchronously, the human operator can also control the slave robot remotely to complete the task by using Leap Motion.

Fig. 3. The robot teleoperation system based on Mixed Reality

A. Leap Motion

In this paper, we adopt Leap Motion as the hand motion tracking device. Developed by Magic LEAP Company, Leap Motion is a USB device with three infrared LEDs and two depth cameras inside. With the help of these equipments, the Leap Motion is able to track the user's finger coordinates and the position of the palms. Besides, the Leap Motion is also used in combination with Head-mounted displays such as Oculus Rift, to obtain a better human-computer interaction performance.

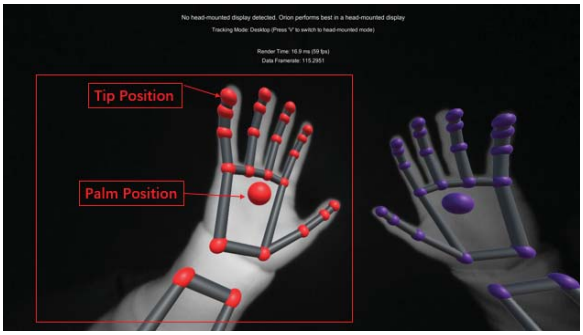


Fig. 4. The interface of Leap Motion

B. Microsoft HoloLens

Released by Microsoft, HoloLens is a novel HMD. Different with common augment reality device, HoloLens allows the user observe the real world environment directly. Besides, the user can also interact with content and information in natural ways, such as gaze, gesture and even voice[18]. Of note, the raw data provided by HoloLens sensors is not available due to the patent protection[4]. Because of the reason above, we have to use the API provided by Microsoft exclusively.

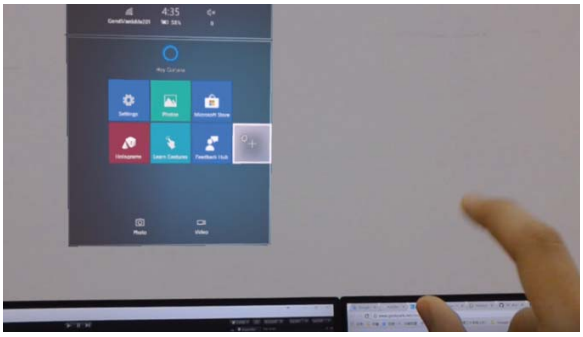


Fig. 5. The interface of HoloLens

C. The slave robot

In our works, a 6-DOFs industrial robot is employed. By using the API provided by the company, we are able to control the robot by sending the end effector position.

Fig. 6. The 6-DOFs robotic arm used in our system

D. Unity

Developed by Unity Technologies, unity is a cross-platform game engine. In order to generate mixed reality scene, the Unity is used in our work. Using assets provided by Leap Motion, we are able to obtain the virtual model of hands in Unity. Furthermore, Unity is allowed to establish communication with HoloLens by using function "Holographic Emulation", and transmit video stream to HoloLens.

978-1-7281-0064-7/19/$31.00 ©2019 IEEE

It's worth mentioning that, to obtain the mixed reality effect, it is essential to set the camera's background color black. Besides, we can also adjust the distance of the virtual objects in "inspector" panel. Besides, the quality of the project should be chosen "fastest", so that the time delay can be reduced to a lower level.

Fig. 7. The interface of Unity

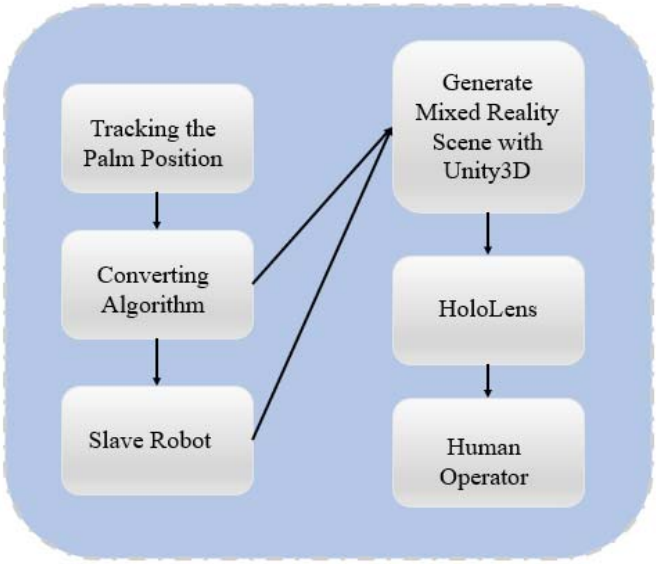


Fig. 8. The implementation steps of the system

E. The Structure of the robot teleoperation system

In this paper, we built up the robot teleoperation system with the Leap Motion and the robotic arm's provided APIs based on C++. We also developed a communication program based on UDP, to transmit the command to the robot in real time. The implementation steps are listed below:

1, Using Leap Motion to capture the user's gestures information.

2, Defining a trigger that the palm is closed or not, start tracking the position of the palm while detecting the palm is closed.

3, Convert the palm's position to the slave robot's Cartesian coordinates with the converting algorithm discussed below.

4, Generate commands applying to the slave robot according to the result of the converting algorithm, then transmit these commands through UDP socket.

The user can observe the slave robot's movement through HoloLens, what's more, the virtual model of the hand generated by Leap Motion is also integrated into the scene for obtaining a better interaction experience.

III. COORDINATE SYSTEM CONVERSION

This section mainly discuss the development of the robot teleoperation system based on mixed reality.

In the process of building this system, it is the key point to transfer the palm's coordinates to the salve robot's Cartesian coordinates, so that the end of salve robot's manipulator can move in pace with the human operator's palm. To address with this problem, we developed a converting algorithm specifically.

A. The coordinate system of Leap Motion

In order to mapping the coordinate values captured from the Leap Motion controller to the robot-defined coordinate system. What's more, the coordinates provided by Leap Motion are in units of millimeters(mm). That means, if the palm's position is identified as $(x, y, z) = [110, -120, 130]$, the position of the palm in real world should be $x = +11cm, y = -12cm, z = 13cm$, accordingly.

The origin of the Leap Motion's coordinate system is defined as the top, center of the device, specifically, the palm's position will be identified as $[0, 0, 0]$, while the user's palm is on the top, center of Leap Motion. Using a right-hand coordinate system, the coordinate system of Leap Motion controller can be shown as Fig.9:

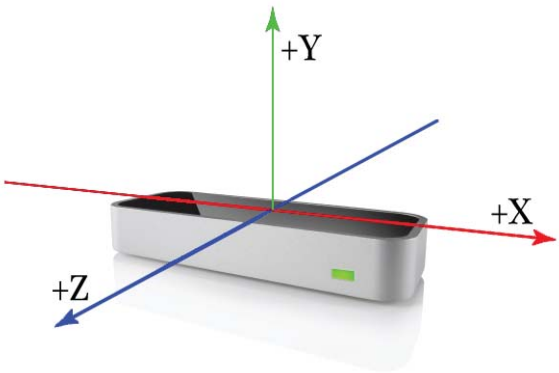


Fig. 9. The coordinate system of Leap Motion Controller[19]

In our work, it is essential to interpret position data for adapting the coordinate system of the slave robot. As the slave robot is working in a 3-dimension space, we adopt all three axes provided by Leap Motion controller.

Commonly, developers are able to use the following equation to make the conversion between these two coordinates:

$$x_{need} = (x_{LM} - LM_{start})\frac{LM_{range}}{need_{range}} + need_{start}. \quad (1)$$

where

$$LM_{range} = LM_{end} - LM_{start}$$
$$need_{range} = need_{end} - need_{start}. \quad (2)$$

What's more, it is necessary to discuss an important function named InteractionBox, which is responsible for defining a cubic area within the Leap Motion controller's field of view.

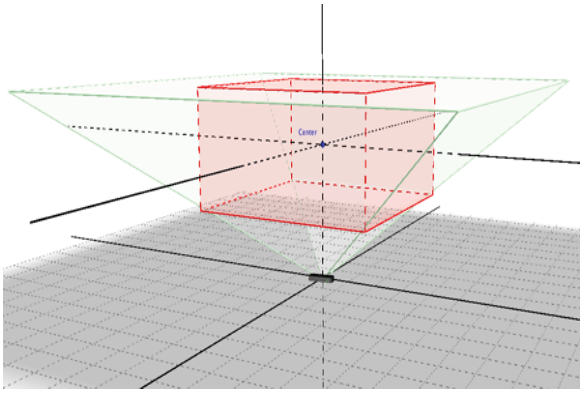


Fig. 10. The InteractionBox of Leap Motion Controller[20]

For obtaining a better performance of tracking, the Leap Motion will change the InteractionBox's size by the field of view as well as the "user's interaction height setting". Nevertheless, this feature will cause a phenomenon that, there may have bias between the normalized coordinates of a point and the normalized coordinates of the point in the real world environment.

In our work, the data captured by Leap Motion is used to control the slave robot directly, the problem mentioned above will trigger a large error and make the slave robot perform poorly. In some particular task, this issue will even bring damage to the objects inside the workspace of the robot.

To address with this problem, we saved a single InteractionBox object and adopted it to normalize all the points in the Leap Motion's field of view.

B. Converting Algorithm

In order to to interpret position data obtained from Leap Motion controller for adapting the coordinate system of the slave robot, we further developed a converting algorithm.

Assuming that, at the initial point, the palm's coordinates in Leap Motion's field of view is $P_{Leap}(x_0^{Leap}, y_0^{Leap}, z_0^{Leap})$, at the same time, it's coordinates of the slave robot is $P_r(x_0^r, y_0^r, z_0^r)$, At time t, the coordinates are $P_{Leap}(x_t^{Leap}, y_t^{Leap}, z_t^{Leap})$, $P_r(x_t^r, y_t^r, z_t^r)$, respectively.

By defining parameters above, the converting relationship can be expressed as below:

$$\begin{cases} x_t^r = x_0^r + C_1(\Delta x_t^{Leap}) \\ y_t^r = y_0^r + C_2(\Delta y_t^{Leap}) \\ z_t^r = z_0^r + C_3(\Delta z_t^{Leap}) \end{cases} \quad (3)$$

where $C = (C_1, C_2, C_3)$ are undetermined parameters, $\Delta P_{Leap} = (\Delta x_t^{Leap}, \Delta y_t^{Leap}, \Delta z_t^{Leap})$ are the change value of the palm coordinates detected by the Leap Motion controller, Consider that, the coordinate system of Leap Motion controller is not match with the coordinate system of the slave robot, they can be expressed as below:

$$\begin{cases} \Delta x_t^{Leap} = z_t^{Leap} - z_0^{Leap} \\ \Delta y_t^{Leap} = x_t^{Leap} - x_0^{Leap} \\ \Delta z_t^{Leap} = y_t^{Leap} - y_0^{Leap} \end{cases} \quad (4)$$

In this experiment, for enhancing the performance of teleoperation, we set $C = (4, 2.7, 40)$, after a list of test.

It should be mentioned that, the position data collected by Leap Motion controller will show up with large fluctuations, which are usually caused negative effects to the control. We also designed an appropriate filter to address with this problem.

$$\begin{cases} \Delta x_t^{Leap} = 0, \Delta x_t^{Leap} \in (-\infty, -6) \bigcup (6, +\infty) \\ \Delta x_t^{Leap} = 0, \Delta x_t^{Leap} \in (-0.6, 0.6) \end{cases} \quad (5)$$

$$\begin{cases} \Delta y_t^{Leap} = 0, \Delta y_t^{Leap} \in (-\infty, -3) \bigcup (6, +\infty) \\ \Delta y_t^{Leap} = -1, \Delta y_t^{Leap} \in (-\infty, -1) \\ \Delta y_t^{Leap} = 1, \Delta y_t^{Leap} \in (1, \infty) \end{cases} \quad (6)$$

$$\begin{cases} \Delta z_t^{Leap} = 0, \Delta z_t^{Leap} \in (-\infty, -5.6) \bigcup (5.6, +\infty) \\ \Delta z_t^{Leap} = 0, \Delta x_t^{Leap} \in (-1, 1) \end{cases} \quad (7)$$

C. The Slave robot workspace

As mentioned above, we are able to control the slave 6-DOFs robot by providing the coordinates of the end-effector. Nevertheless, in some particular situations, the coordinates we send is not in the robot's workspace. To protect the robot as well as objects around it, it is necessary for us to figure out the slave robot's workspace and limit points in a reasonable threshold value.

Under the assumption of knowing the slave robot's DH parameters, we can calculate the homogeneous transformation matrix with the following equation:

$$^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\cos\alpha_i & \alpha_i\cos\theta_i \\ \cos\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\cos\alpha_i & \alpha_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

We are able to calculate the the coordinates of the slave robot's base by using the coordinates of the end-effector:

$$^nX_0 = {}^0A_1\,{}^1A_2 ... {}^{n-1}A_n \cdot X_n. \quad (9)$$

Where n represents the DOF of the slave robot, while $X_0 = [x_0, y_0, z_0, 1]$,$^nX_0 = [x_n, y_n, z_n, n]$represents the coordinates of the slave robot's base, the coordinates of the end-effector, respectively.

Combining with the slave robot's range of movement shown as Table.1:

TABLE I
THE SLAVE ROBOT'S RANGE OF MOVEMENT

Joint	range of movement
J1	$\pm 0.02mm$
J2	$-160°/+10°$
J3	$-25°/+235°$
J4	$\pm 180°$
J5	$\pm 105°$
J6	$\pm 360°$

It should be mentioned that, the slave robot has a software protection mechanism to protect it from moving outside the specified workspace. While the software protection mechanism is triggered, the robot will enter emergency stop state immediately, and the process of teleoperation will be interrupted. To address with this problem, we further develop a workspace restraint for our teleoperation system, which can be expressed as follow:

$$\begin{cases} z_t^{robot} = 223.399, & z_t^{robot} \in (223.399, \infty) \\ z_t^{robot} = 164, & z_t^{robot} \in (-\infty, 164) \end{cases} \tag{10}$$

$$\begin{cases} x_t^{robot} = 365, & x_t^{robot} \in (365, \infty) \\ x_t^{robot} = 224, & x_t^{robot} \in (-\infty, 224) \end{cases} \tag{11}$$

$$\begin{cases} y_t^{robot} = 60, & y_t^{robot} \in (60, \infty) \\ y_t^{robot} = -60, & y_t^{robot} \in (-\infty, -60) \end{cases} \tag{12}$$

After coordinate system conversion done by converting algorithm discussed above, we can assume that the coordinate of the palm and the end-effector of the slave robot is matched approximately. What's more, with the help of workspace restraint, we can make sure that the robot's software protection mechanism won't be triggered while the human operator's manipulation is non-compliance.

IV. EXPERIMENTAL RESULTS

In this paper, to verify the validity of the teleoperation system, we designed a specific task, in which the human operator teleoperate the slave robot to write a simple Chinese character.

As mentioned above, the robot teleoperation system is consisted with 4 parts: the human operator, a Leap Motion controller and a slave robot, as shown in Fig.11:

With the help of Leap Motion controller, the human operator is able to control the slave robot remotely. At the same time, the operator can also obtain a real-time visual through HoloLens. The mixed reality scene is generated by Unity3D and HoloLens, to offer a better teleoperating performance for the operator.

During the teleoperational process, the operator can obtain a mixed reality scene as shown in Fig.12:

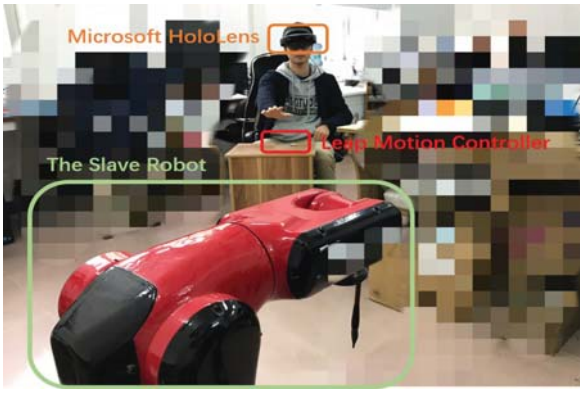


Fig. 11. The operator teleoperates the slave robot

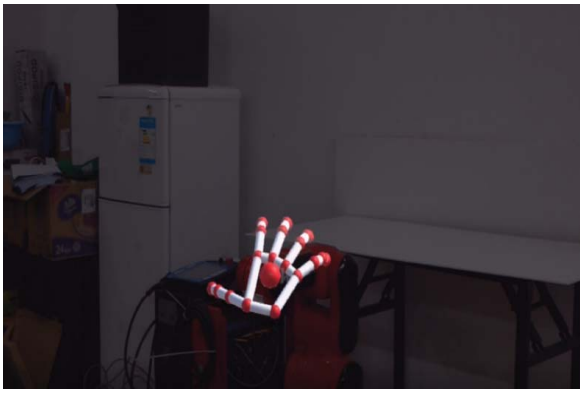


Fig. 12. The Mixed Reality Scene observed from HoloLens

Through HoloLens, the operator is able to observe the slave robot and a virtual model of the hand, which make the process of teleoperation more intuitive. Besides, with the help of the mixed reality visual feedback, the operator is able to move the hand refer to the virtual model, the difficulty of controlling the slave robot is reduced.

In this task, we set the virtual model of hand's position as $P(x = 0, y = -0.3, z = 2)$, that means, in the operator's view, the virtual model is 2 meters far away.

During the task, we need to define a trigger for determining if the teleoperation is running or not, so that the operator can start or stop the process at any time he/she want.

In our experiments, we define making a fist as the trigger. If the Leap Motion detected that the operator has made a fist, the program will start recording the position of the palm, and generating commands to control the slave robot.

The task we designed for this section is that, by using the robot teleoperation system, writing a Chinese character "Wang" with the 6-DOFs slave robot, as shown in fig.13.

In the experiments, by using the robot teleoperation system based on mixed reality, the human operator is able to finish the task with the slave robot. Besides, the operator can obtain a real-time visual feedback made with mixed reality scene.

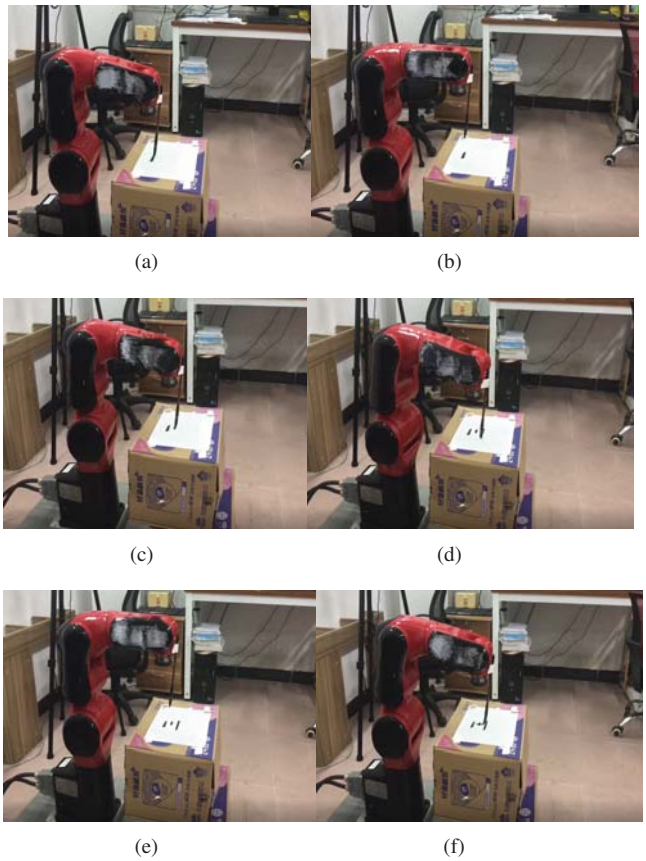
During the experiment, with the help of the visual feedback transmitted by HoloLens, the operator is able to adjust the operation timely to obtain a better performance of control, and the workspace restraint also limits the movement of

978-1-7281-0064-7/19/$31.00 ©2019 IEEE

the slave robot within a specified space.

V. CONCLUSIONS

In this paper, we mainly developed a robot teleoperation system based on mixed reality. The teleoperation system was consisted of four parts: the human operator, a Leap Motion controller, a Microsoft HoloLens, and a slave robotic arm. Furthermore, to improve the performance of the teleoperation, we also presented a converting algorithm to convert the coordinate system of Leap Motion to the coordinate system of the slave robot. By employing the workspace restraint, the slave robot was able to rise superior to the operator's misoperation. Besides, the teleoperation system also offered a real-time visual feedback to the operator, by generating mixed reality scene and displaying with the Microsoft HoloLens Head-mounted display. We also designed a simple task, in which the human operator teleoperated a 6-DOFs robotic arm to write a simple Chinese character. The experiments have shown that, with the proposed teleoperation system, the process of teleoperation could be effective.



(a) (b) (c) (d) (e) (f)

Fig. 13. Writing a Chinese character with the slave robot

REFERENCES

[1] C. Yang, J. Luo, C. Liu, M. Li, and S.-L. Dai, "Haptics electromyogrphy perception and learning enhanced intelligence for teleoperated robot," *IEEE Transactions on Automation Science and Engineering*, 2018.

[2] J. Cui, S. Tosunoglu, R. Roberts, C. Moore, and D. W. Repperger, "A review of teleoperation system control," in *Proceedings of the Florida Conference on Recent Advances in Robotics (FCRAR)*, 2003, pp. 1–12.

[3] S. Lichiardopol, "A survey on teleoperation," *Technische Universitat Eindhoven, DCT report*, 2007.

[4] L.-Y. Hu, X. P. Liu, and G.-P. Liu, "The wave-variable teleoperator with improved trajectory tracking," in *Control and Automation (ICCA), 2010 8th IEEE International Conference on*. IEEE, 2010, pp. 322–327.

[5] J. Chen, M. Glover, C. Li, and C. Yang, "Development of a user experience enhanced teleoperation approach," in *Advanced Robotics and Mechatronics (ICARM), International Conference on*. IEEE, 2016, pp. 171–177.

[6] S. Livatino, F. Banno, and G. Muscato, "3d integration of robot vision and laser data with semi-automatic calibration in augmented reality stereoscopic visual interface," *environments*, vol. 15, no. 18, p. 19, 2012.

[7] Y. Sung and K. Cho, "Collaborative programming by demonstration in a virtual environment," *IEEE Intelligent Systems*, vol. 27, no. 2, pp. 14–17, 2012.

[8] M. Lorenz, M. Busch, L. Rentzos, M. Tscheligi, P. Klimant, and P. Fröhlich, "I'm there! the influence of virtual reality and mixed reality environments combined with two different navigation methods on presence," in *Virtual Reality (VR), 2015 IEEE*. IEEE, 2015, pp. 223–224.

[9] R. Kumar, T. Oskiper, O. Naroditsky, S. Samarasekera, Z. Zhu, and J. Kim, "System and method for generating a mixed reality environment," Mar. 21 2017, uS Patent 9,600,067.

[10] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[11] "Microsoft hololens," https://www.windowscentral.com/microsoft-hololens.

[12] H. F. Vargas and O. A. Vivas, "Gesture recognition system for surgical robots manipulation," in *Symposium on image, signal processing and artificial vision*, 2014, pp. 1–5.

[13] E. GRADY, "Engineer thursday - tearing apart the leap motion," https://www.sparkfun.com/news/1190.

[14] J. Luo, C. Yang, N. Wang, and M. Wang, "Enhanced teleoperation performance using hybrid control and virtual fixture," *International Journal of Systems Science*, pp. 1–12, 2019.

[15] F. Hernoux, R. Bearee, L. Gajny, E. Nyiri, J. Bancalin, and O. Gibaru, "Leap motion pour la capture de mouvement 3d par spline l1," in *Journées du Groupe de Travail en Modélisation Géométrique 2013, Marseille*, 2013.

[16] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *ISR/robotik 2014; 41st international symposium on robotics; proceedings of*. VDE, 2014, pp. 1–7.

[17] L. Peppoloni, F. Brizzi, C. A. Avizzano, and E. Ruffaldi, "Immersive ros-integrated framework for robot teleoperation," in *3D User Interfaces (3DUI), 2015 IEEE Symposium on*. IEEE, 2015, pp. 177–178.

[18] "Microsoft hololens," https://www.microsoft.com/en-us/hololens.

[19] "Leap concepts, coordinate system," https://developer.leapmotion.com/documentation/v4/concepts.html.

[20] "Leap coordinate mapping," http://www.xionggf.com/articles/vr/leapmotion/leapmotion_coordination_and_transform.html.