A Markerless Human–Robot Interface Using Particle Filter and Kalman Filter for Dual Robots

Guanglong Du and Ping Zhang

***Abstract*—A more natural means of communicating complex movements to a robot manipulator is where the manipulator copies the movements of human hands. This paper presents a markerless human–robot interface that incorporates Kalman filters (KFs) and particle filters (PFs) to track the posture of human hands. This method allows one operator to control dual robot manipulators by using his/her double hands without any contact devices or markers. The algorithm employs Leap Motion to determine the orientation and the position of the human hands. Although the position and the orientation of the hands can be obtained from the sensor, the measurement errors increase over time due to the noise of the devices and the tracking error. The PFs and KFs are used to estimate the position and the orientation of the human hand. Due to the limitations of the perception and the motor, a human operator cannot accomplish high-precision manipulation without any assistance. An adaptive multispace transformation is employed to assist the operator to improve the accuracy and reliability in determining the posture of the manipulator. The greatest advantage of this method is that the posture of the human hands can be estimated accurately and steadily without any assistant markers. The human-manipulator interface system was experimentally verified in a laboratory, and the results indicate that such an interface can successfully control dual robot manipulators even if the operator is not an expert.**

***Index Terms*—Human–robot interface, Kalman filter (KF), markerless, particle filter (PF).**

I. Introduction

HUMAN intelligence is necessary when a robot is in a dynamic and unstructured environment. Therefore, a human–robot interface is necessary when the robot is in such an environment. There are some human–robot interfaces [1], such as joysticks [2]. Compared with using contacting mechanical devices, tracking the human movements to control the robot using inertial sensors, angle sensors, or exoskeletal systems [3] is more natural. However, the contacting ways may hinder the human motion.

Noncontacting ways such as vision-based techniques may not hinder human movements. The methods in [4] placed the physical markers on the body part to track the human hand to control the remote robot. Because of the possibility of occlusion, multiple markers are attached on the body, and multiple cameras are used for detecting the markers normally.

The markerless tracking way is a real noncontacting human–robot interface. Compared with the marker-based method, the markerless way does not suffer from marker identification and marker occlusion [5]. Therefore, the markerless way seems a better method for robot manipulation. Markerless tracking methods [6], [7] tracked the human arm by computing the movements later or capturing the image of the human arm. The operator could control the manipulator continuously by capturing the continuous movements of the human arm. The orientation and the position of the hand arm are calculated immediately, which allows the operator to complete the manipulation in a natural way. However, there are many limitations on using the markerless method in robot teleoperation. The markerless methods in [8] captured the 2-D images of the human hand to control the remote robot; therefore, it is hard to be extended for extracting 3-D movements of the human hand. The robot can be controlled continuously with the 3-D orientation and position of the human hand. How to obtain the 3-D orientation and position of human hand is a challenge [6]. There are some limited studies on markerless tracking, such as that in [9]. They used the hand gestures of the operator to control the remote robot. These methods contain only a few different commands such as "up," "down," "go," and "stop," which are applicable for a mobile robot [10]. However, it is not possible for a manipulator to complete the manipulation in 3-D space by using natural and flexible movements. When an operator wants a manipulator to move down, up, or forward, he has to think which gesture to be used. Therefore, it is better for a human operator to focus on the manipulation task instead of thinking which gesture should be used to control the robot to move up or down. Kofman *et al.* [11] tracked the hand-arm movements in the 3-D space so that it could allow the operator to focus on the global task naturally. However, the method needs the operator doing the gestures in the dark background, with the hands being higher than the shoulder. When the operator is in a complex environment, the method cannot get a precise gesture since it suffers from the lighting effects, such as being too bright or too dark. Moreover, the operator cannot work in the chill weather with an unclothed arm. Du *et al.* [12] used a Kinect to track the human hand of the operator for controlling dual robot manipulators. This method obtains the orientation through tracking the thumb and the index

Manuscript received April 4, 2014; revised July 7, 2014 and August 11, 2014; accepted September 16, 2014. Date of publication October 8, 2014; date of current version March 6, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61403145, in part by the China Postdoctoral Science Foundation under Grant 2014M550436, and in part by the Fundamental Research Funds for the Central Universities under Grant 2014ZM0039. *(Corresponding author: Ping Zhang.)*

G. Du is with the School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: medgl@scut.edu.cn).

P. Zhang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: pzhang@scut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2014.2362095

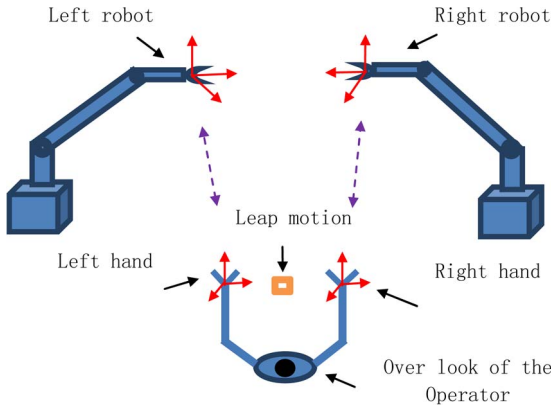

Fig. 1. Noninvasive robot teleoperation system.

finger. However, this method is easily affected by the occlusion. In addition, the method has low precision.

In order to obtain more accurate and reliable estimation from the sensors, the position can be integrated with the orientation, so that the drawbacks of each estimation are compensated [13]. Kalman filters (KFs) [14] or particle filters (PFs) [15] are widely used to integrate the orientation and the position. KFs can estimate the states of linear Gaussian state-space models, and PFs can deal with highly nonlinear models. To decrease the computational complexity, the linear state can be estimated by KFs, whereas the remaining part is estimated by PFs [16]. The proposed system uses Leap Motion (LM) [17] to locate the two hands of the operator to control the dual robots (see Fig. 1). This paper proposes KFs to estimate the orientation of the human hands, PFs to estimate the position, and an adaptive multispace transformation (AMT) method to improve the accuracy of manipulation. Experimental results to validate the proposed methods are also presented.

The remainder of this paper is organized as follows. In Section II, an overview of this paper is described. The human-hand tracking system is detailed in Section III. Orientation estimation and position estimation are presented in Section IV and Section V, respectively. Section VI describes the method of AMT. Experiments and results are presented in Section VII. Discussions are detailed in Section VIII, followed by concluding remarks in Section IX.

II. Overview

Fig. 1 shows the structure of the presented system, and it is the top view of the operator. LM is placed on the table, and the operator waves his hands to control the dual robots. The operator just considers that the dual robots are his hands so that controlling the dual robots is similar to controlling his own hands. Fig. 2 shows the process of the proposed algorithm. In the proposed system, although LM can provide the posture of the human hands, the posture data contains noise, even the tracking failure data. In order to eliminate the influence of the noise and tracking failure data, KFs and PFs are used to estimate the orientation and the position of the human from the measured data. The system uses the estimation data to control the robot instead of the measured data. Moreover, humans have inherent perceptive limitations (such as perception of distance) and motor limitations (such as physiological tremor), which prevent them from operating precisely and smoothly enough for certain tasks. In this paper, AMT is presented to improve the accuracy of the manipulations.

III. Hand Tracking

A. Hand Coordinate System

Tracking of the operator hand motion at the local site is carried out by a hand tracking system consisting of LM, which is detailed in [18]. In the markerless hand manipulator interface, the 3-D position and orientation of the centroid on the hand are used to control the position and orientation of the robot end-effector. The 3-D frames of the tracking system are shown in Fig. 3. The LM frame $X_LY_LZ_L$ is defined by a translation of the world-fixed frame $X_WY_WZ_W$. The hand frame $X_HY_HZ_H$ is defined by Leap motion translation of the LM frame $X_LY_LZ_L$. The center of the human hand is defined as the centroid on the hand. X_H is collinear with the middle finger, and it points out, whereasY_H is perpendicular to the back of a hand and it points upward. Z_H is the multiplication cross of X_H and Y_H. The position of the hand is the translation of the frame $X_HY_HZ_H$ to the frame $X_LY_LZ_L$.

The orientation of the hand is defined as the yaw–pitch–roll frame rotation angles: yaw, pitch, and roll represent the rotation ϕ, θ, and ψ between the frame $X_LY_LZ_L$ and the frame $X_HY_HZ_H$ about the X, Y, Z-axes. The LM can obtain the position, velocity, rotation angles, and angular velocity of the human hand with respect to the frame $X_LY_LZ_L$.

B. FQA

The factored quaternion algorithm (FQA) presented in [19] is adopted to estimate the orientation of the rigid body. This algorithm is only applicable to static or slow-moving rigid bodies. However, the KF fusion algorithm using angular rate information can make the FQA adapt to estimate the orientation of dynamic bodies (slow or fast moving) in situations with relatively large linear acceleration.

The LM measures the orientation (roll, pitch, and yaw) of the human hand in its own frame $X_LY_LZ_L$. Rotation ϕ around the X_L-axis represents roll. Rotation θ around the Y_L-axis represents pitch, and rotation ψ around the Z_L-axis represents yaw. According to Euler's theorem on finite rotations, the conversion from Euler angles to quaternion is

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)+\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)-\cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right)+\sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right)-\sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{1}$$

where q_0, q_1, q_2, and q_3 are the quaternion components that satisfy

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{2}$$

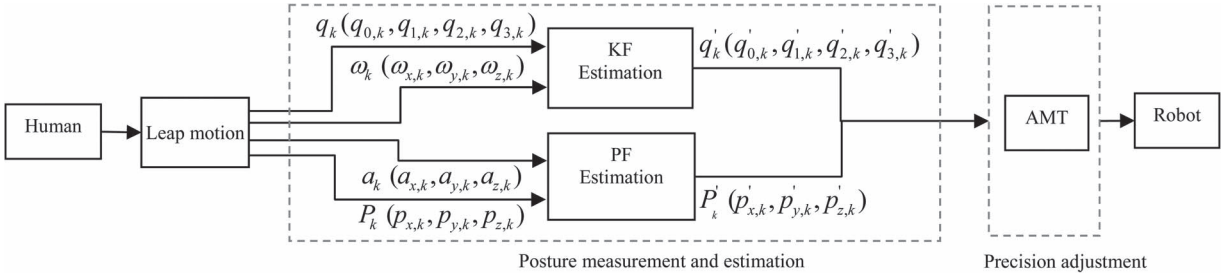

Fig. 2. Human–robot interface system process.

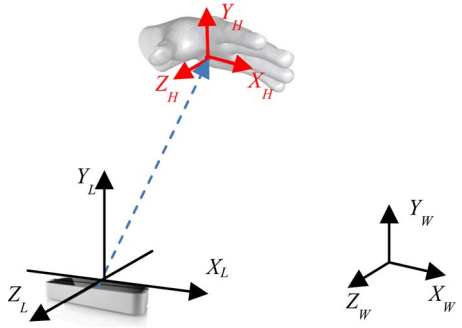

Fig. 3. Coordinate system of LM.

IV. Orientation Estimation

The KF, a recursive stochastic algorithm, is used for the estimation of the state at time k by using the state at time $k-1$ [20]. The KF is used to estimate the state x of a human hand from a set of noisy and incomplete measurements [14]. The differential equation of the quaternion q with respect to time t is

$$\begin{bmatrix} \frac{\partial q_0}{\partial t} \\ \frac{\partial q_1}{\partial t} \\ \frac{\partial q_2}{\partial t} \\ \frac{\partial q_3}{\partial t} \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \frac{\omega_x}{2} \\ \frac{\omega_y}{2} \\ \frac{\omega_z}{2} \end{bmatrix} \tag{3}$$

where ω_x, ω_y, ω_z are the angular velocity components of a human hand in x_s, y_s, z_s-axes. Since the state x_k includes the quaternion states and the angular velocities, x_k has the following format:

$$x_k = [q_{0,k}\ q_{1,k}\ q_{2,k}\ q_{3,k}\ \omega_{x,k}\ \omega_{y,k}\ \omega_{z,k}] \tag{4}$$

where $q_{0,k}$, $q_{1,k}$, $q_{2,k}$, $q_{3,k}$, $\omega_{x,k}$, $\omega_{y,k}$, and $\omega_{z,k}$ are the quaternion states and the angular velocities at time k.

The quaternion components of each particle can be calculated from the angular velocity measurements using the following:

$$\begin{bmatrix} q_{0k} \\ q_{1k} \\ q_{2k} \\ q_{3k} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & -\omega_{x,k-1} \cdot t & -\omega_{y,k-1} \cdot t & -\omega_{z,k-1} \cdot t \\ \omega_{x,k-1} \cdot t & 2 & \omega_{z,k-1} \cdot t & -\omega_{y,k-1} \cdot t \\ \omega_{y,k-1} \cdot t & -\omega_{z,k-1} \cdot t & 2 & \omega_{x,k-1} \cdot t \\ \omega_{z,k-1} \cdot t & \omega_{y,k-1} \cdot t & -\omega_{x,k-1} \cdot t & 2 \end{bmatrix} \cdot \begin{bmatrix} q_{0k-1} \\ q_{1k-1} \\ q_{2k-1} \\ q_{3k-1} \end{bmatrix} \tag{5}$$

where the sample time is t.

From (3) and (5), the state-transition matrix is

$$\Phi_k = \begin{bmatrix} 1 & 0 & 0 & 0 & -q_{1,k} \cdot \Delta t/2 & -q_{2,k} \cdot \Delta t/2 & -q_{3,k} \cdot \Delta t/2 \\ 0 & 1 & 0 & 0 & q_{0,k} \cdot \Delta t/2 & q_{3,k} \cdot \Delta t/2 & q_{2,k} \cdot \Delta t/2 \\ 0 & 0 & 1 & 0 & q_{3,k} \cdot \Delta t/2 & q_{0,k} \cdot \Delta t/2 & -q_{1,k} \cdot \Delta t/2 \\ 0 & 0 & 0 & 1 & -q_{2,k} \cdot \Delta t/2 & q_{1,k} \cdot \Delta t/2 & q_{0,k} \cdot \Delta t/2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where Δt is sampling time. Let system input matrix Γ_{ori} is a zero matrix because there is no control inputs. We use angular velocities to estimate the quaternion states; therefore, the process noise vector is

$$w_k = [0\ 0\ 0\ 0\ w_x\ w_y\ w_z]^T \tag{7}$$

where w_x, w_y, and w_z are the process noise of the angular velocity. Because we use LM to measure angular velocities, the observation matrix H is

$$H_{\text{ori}} = [0^{n \times p}\ I^{n \times n}] \tag{8}$$

where n is the number of angular velocities vector and p is the number of the quaternion. The determined quaternion q_k at time k should be normalized as

$$q_k = \left[\frac{q_{0,k}}{\mathbf{M}}\ \frac{q_{1,k}}{\mathbf{M}}\ \frac{q_{2,k}}{\mathbf{M}}\ \frac{q_{3,k}}{\mathbf{M}} \right]$$
$$\mathbf{M} = \sqrt{q_{0,k}^2 + q_{1,k}^2 + q_{2,k}^2 + q_{3,k}^2}\,. \tag{9}$$

V. Position Estimation

A. PF

The PF is a suboptimal resolution, which estimates the true posterior using a finite number of random state samples among their corresponding normalized weights [13], [21]. Thus, at time t_k, the posterior density approximation is

$$p(x_k | z_{1:k}, u_{0:k-1}) \approx \sum_{i=1}^{N} \omega_k^i \delta\left(x_k - x_k^i\right) \tag{10}$$

where $\delta(\cdot)$ is the Dirac delta function, N is the number of samples, ω_k^i is the normalized weight of the ith particle, and x_k^i is the ith particle.

For derivation of the PF state estimation algorithm, the posterior density up to time t_k is

$$p(x_{0:k}|z_{1:k},u_{0:k-1}) = \frac{p(z_k|x_k)\cdot p(x_k|x_{k-1},u_{k-1})\cdot p(x_{0:k-1}|z_{1:k-1},u_{0:k-2})}{p(z_k|z_{1:k},u_{0:k-1})}. \quad (11)$$

Since sampling from the posterior density is frequently difficult, an importance sampling technique [22] is employed. If target density (posterior density in this case) can be assessed at any juncture, but sampling remains difficult, samples may be obtained from a recognized normalized probability density $[r(x)]$, which is called importance density. In order to counteract for variances between the target and the importance densities, normalized weights defined as the ratios of the two densities are given to each particle [23]. Up to time t_k, the discrete posterior density approximation is described as

$$p(x_{0:k}|z_{1:k},u_{0:k-1}) \approx \sum_{i=1}^{N} \omega_k^i \delta\left(x_{0:k} - x_{0:k}^i\right). \quad (12)$$

When the importance density is calculated from prior as

$$r(x_k|x_{k-1},z_k,u_{k-1}) = p(x_k|x_{k-1},u_{k-1}) \quad (13)$$

then the normalized weight can be written as

$$\omega_k^i \propto \omega_{k-1}^i \cdot p\left(z_k|x_k^i\right). \quad (14)$$

This type of a PF is problematic in that, when k is high, only one particle will possess higher weights (nearing unity), whereas the weights of all remaining particles will be negligible (approaching to zero). This phenomenon, referred to as the degeneracy problem, is unfavorable since the weighted particles fail to accurately reflect the real posterior density. To circumvent this issue, resampling of particles on the basis of their weights should be performed. After resampling, all particles are assigned the same weight; the weights at time t_{k-1} are identical $(\omega_k^i = 1/N)$. Resampling reduces the sample numbers from the lower weights by drawing more higher weight samples.

B. Position Estimation Using PF

This proposed method estimates the object's position by using a PF, whose state is set as $[x_{\mathrm{PF}} = (p_x, p_y, p_z)]$. The position measurements can be used as the expected value and the position calculation of each position particle can be used to determine the weight of the particle. However, differences for each position state in the position measurements and position calculation are not only due to the position state but also due to noise from sensor errors. Therefore, instead of using instantaneous position difference, the summation of the position difference is used to update the weight of each particle. Moreover, the combination of KF orientation estimations of each particle, and the acceleration measurements is used to reduce the effect of the measurement noise, instead of using direct position [16].

For a time period (ΔT_s, where subscript s is the sth position iteration, $s = 1, 2, \ldots$), the summation of the position difference is used to obtain the weights rather than the immediate position difference at time t_k. Then, the accumulated position difference from the estimated and calculated values for the ith particle is used in the likelihood calculation as in the following:

$$\mathrm{PE}_s^i = \sum_{k=(s-1)\cdot M_S+1}^{M_s\cdot s} \sum_{j=(x,y,z)} \left[P_{p-j,k}^i - \left(P_{p-j,k-1}^i + \frac{1}{2}R_k a_{j,k}^i t^2\right)\right]^2 \quad (15)$$

where $M_s = \Delta T_s/t$, and $P_{p-j,k}^i$ is the position states at time t_k. R_k is the orientation state of the KF at time t_k. $a_{j,k-1}^i$ is the speed of the human hand at time k. For a given particle, the lower the PE_s^i is, the greater the likelihood is. Based on the PE_s^i values, the weight of each particle is recalculated for every ΔT_s period.

Since the minimum accumulated error leads to have the highest likelihood of the correct position, $\arg\min(\mathrm{PE}_s^i)$ is defined as the most probable value. Moreover, the weight is calculated based on PE_s^i; the normalized weight is expressed as

$$\omega_s^i \propto \exp\left(\frac{-\left(\mathrm{PE}_s^i - \arg\min\left(\mathrm{PE}_s^i\right)\right)^2}{2\times\left(\sigma\left(\mathrm{PE}_s^i\right)\right)^2}\right). \quad (16)$$

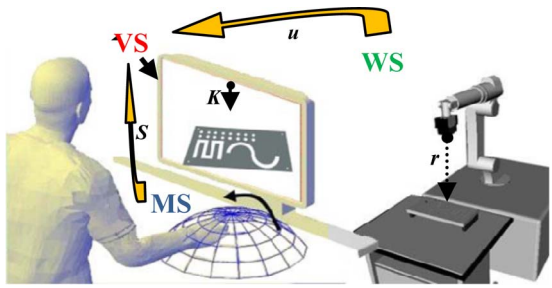

Fig. 4. Representation of the human–interface–robot spaces.

VI. AMT

Humans have inherent perceptive limitations (such as perception of distance) and motor limitations (such as physiological tremor), which prevent them from operating precisely and smoothly enough for certain tasks [24]. For improving the visual and motor performance of teleoperation interfaces, we applied the modified version of AMT[24]. In the proposed method, an interface that introduces two scaling processes links the human operator working space (WS) to the robot WS (see Fig. 4). The first change scales the movement produced by the human operator. Another change of scale is applied between the virtual unit vector K of the central axis of the robot end-effector and the robot movements. Such changes of scale modify the robot speed so that it can improve the performance.

The scaling vector S is used to relate the actions of the human hand in master space (MS) and the movement of the virtual unit vector K in the visual space (VS). Another scaling variable u is used to relate the VS space to the robot WS. S and u are two functions of the distance r between robot end-effector and the target. When $S < 1$, it decelerates the movement of K. Instead, it accelerates the movement of K while $S > 1$.

As the movement of the virtual position in VS is affected by vector S, let $S = [S_{\text{pos}}, S_{\text{ori}}]$, where S_{pos} is the scaling vector of the position, and S_{ori} is the scaling variable of the orientation. The Euler angular velocity in MS is $\dot{O}_M$, and $\dot{O}_V$ is the angular velocity in VS. Then,

$$\dot{O}_V = \begin{cases} S_{\text{ori}} \cdot \dot{O}_M, & \dot{O}_M \le \delta_{\text{ori}} \\ 0, & \dot{O}_M > \delta_{\text{ori}} \end{cases} \tag{17}$$

where δ_{ori} is the threshold value. Assume that P_M is the position vector of the hand movement in MS, and P_V is the position of the vector K in VS, then the mapping vector of $\dot{P}_M$ in the VS is $\dot{P}'_M$. Defined that $K_\perp$ is a vector that is perpendicular to K. Let $S_{\text{pos}} = [s_k \; s_{k_\perp}]$, where s_k is the scaling value in the K direction, and $s_{k_\perp}$ is the scaling value in the $K_\perp$ direction. In addition, $K_\perp$, K, and $\dot{P}'_M$ are coplanar. Therefore, we have

$$p_k = \begin{cases} s_k \cdot \left(\left| \dot{P}'_M \right| \cdot \cos\theta \right) \cdot K = s_k \cdot K \cdot P'_M \cdot K & S_k \le \delta_k \\ 0 & S_k > \delta_k \end{cases}$$

$$p_{k_\perp} = \begin{cases} s_{k_\perp} \cdot \left(\left| \dot{P}'_M \right| \cdot \sin\theta \right) \cdot K_\perp \\ \quad = s_{k_\perp} \cdot K_\perp \cdot P'_M \cdot K_\perp & S_{k_\perp} \le \delta_{k_\perp} \\ 0 & S_{k_\perp} > \delta_{k_\perp} \end{cases} \tag{18}$$

where δ_k and $\delta_{k_\perp}$ are the threshold values. Then, the speed $\dot{P}_V$ of the vector K is given as

$$\dot{P}_v = p_k + p_{k_\perp}. \tag{19}$$

When $s_k = s_{k_\perp}$, the speed $\dot{P}_V$ results in $\dot{P}_v = s_k P'_M$. While $s_k < s_{k_\perp}$, it requires greater precision in the direction of the central axis of end-effector. The direction that is perpendicular to the central axis requires greater precision when $s_k > s_{k_\perp}$. In this paper, we use $s_k > s_{k_\perp}$ so that the operator can easily center the axis of the hole and insert the peg into the hole quickly.

S_{ori} and S_{pos} are the functions of distance, which are defined as follows:

$$\begin{cases} S_{\text{ori}}(r) = \log(r) + C_1 \\ S_{k_\perp}(r) = \log(r) + C_2 \\ S_k(r) = \sqrt{r} + C_3 \end{cases} \tag{20}$$

where C_1, C_2, and C_3 are constants.

The speed $\dot{P}_W$ can be given as

$$\dot{P}_W = \left(\dot{P}_V + \dot{p} - (P_W - P_C)\frac{du}{dr}\dot{r} \right) \frac{1}{u} \tag{21}$$

where P_C is the position of the zoom center in VS, P_W is the position of the end-effector in WS, and $\dot{p}$ is the panning speed. In order to keep both the target and the virtual vector within the VS, $\dot{p}$ is given as

$$\dot{P} = s_{\text{pos}}\dot{P}_M + (P_W - P_C)\frac{du}{dr}\dot{r}. \tag{22}$$

Since u is a function of distance r, the function $u(r)$ is defined as

$$u(r) = \frac{C_1}{r} + C_2 \tag{23}$$

where C_1 and C_2 are constants.

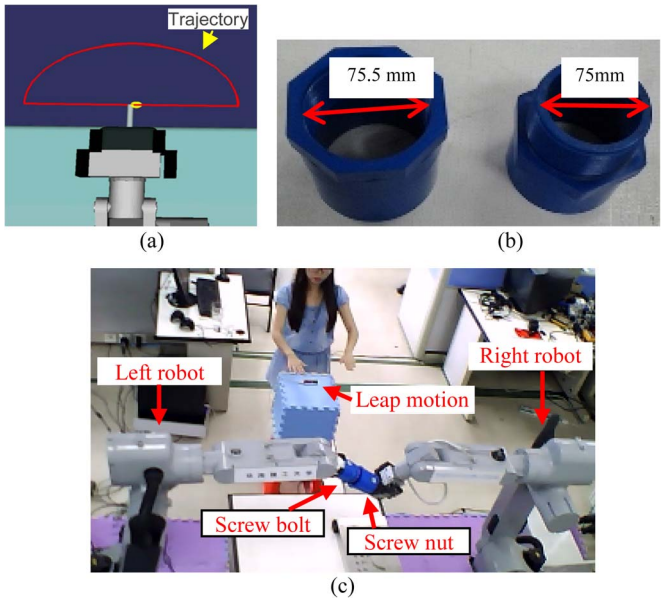

Fig. 5. Environments of experiments. (a) Experiment 1. (b) The screw nut and the screw bolt. (c) Experiment 2.

TABLE I
DH PARAMETERS FOR THE ROBOT (a: LINK LENGTH. α: LINK TWIST. d: LINK OFFSET. θ: JOINT ANGLE)

DH / Joint	a (mm)	α (rad)	d mm)	θ (rad)
1	150	-π/2	250	0
2	570	-π	0	-π/2
3	150	π/2	0	0
4	0	-π/2	650	0
5	0	-π/2	0	-π/2
6	0	0	-200	0

VII. EXPERIMENTS

A. Environment of Experiment

A series of experiments were carried out to verify the proposed method. In Experiment 1, a simulation experiment was designed to evaluate the accuracy of the human–robot interface [see Fig. 5(a)]. In Experiment 2, a series of screwing bolt was carried out to compare our method with methods [11], [12] in efficiency of the manipulation [see Fig. 5(c)]. The three methods controlled the position and orientation of the robot end-effector directly to screw the bolt into the nut. Joint angles can be achieved by the solution of robot inverse kinematics. Given a posture of the robot end-effector, the angle of each joint can be achieved by the robot inverse kinematics [25]. Since the operating space is smaller than the WS of the robot, the system will stop when the operator clenches his fist so that the operator can reset his/her hand to the center over the LM. Fig. 5 shows the environment of the experiment.

Two GOOGOL GRB3016 robots were used in our experiments. Table I lists the nominal robot link parameters of the robots. Two robot manipulators were placed at the distance of 205.3 cm. An LM was placed in front of the robots, and the operator stood in front of the LM. The LM was sampled at 30 Hz.

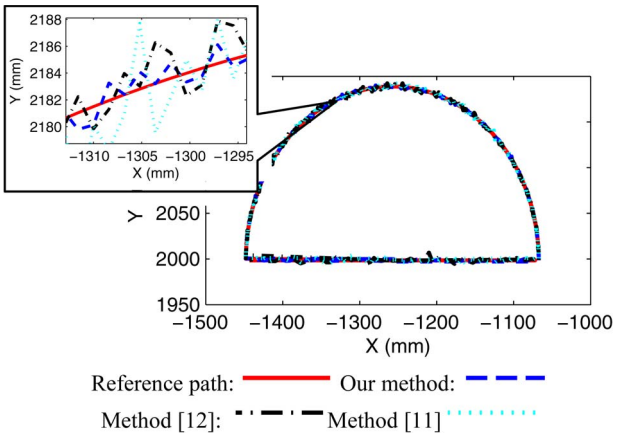

Fig. 6. Tracking results.

TABLE II
TRACKING ERRORS OF THE THREE METHODS

	Minimum (mm)	Mean (mm)	Maximum (mm)	Time (s)
Our method	0.16	1.56	3.32	148
Method [12]	0.23	3.87	6.89	176
Method [11]	0.32	4.65	8.34	203

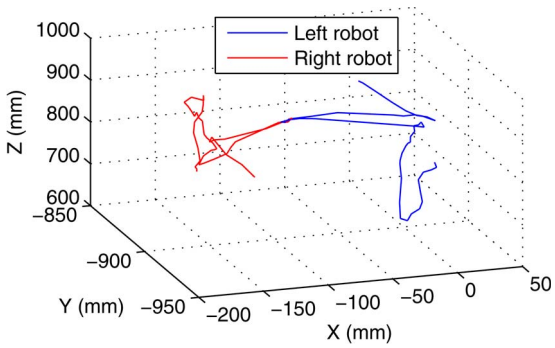

Fig. 7. Robots paths.

In Experiment 1, the radius of the trajectory was 190 mm. In Experiment 2, the diameter of the nut was 75.5 mm, and the diameter of the bolt was 75 mm [see Fig. 5(b)]. The length of the nut was 91 mm, and that of bolt was 83 mm. In each task, the operator needed to control the robots and screwed the bolt into the nut. The matrices R and Q for KF can be determined by the adaptive method described in [27] and [28].

B. Results of Experiment

A comparison experiment was completed in Experiment 1. Fig. 6 shows the tracking results of the three methods. The solid line is the reference path. The dashed line is the tracking path of our method. The dash-dotted line is the tracking path of method [12]. The dotted line is the tracking path of method [11]. Table II presents the tracking errors of the three methods. The mean error of our method was less 2.31 mm, which is 3.09 mm than that of the method in[12] and that of the method in[11].

Fig. 7 shows the 3-D path of two robot end-effectors during the screwing in Experiment 2.The whole time of each test was about 240 s. Figs. 8 and 9 show the results of tracking the robot end-effectors in the period of adjusting the position and orientation (about 35 s). The period of screwing bolt was ignored since the position and the orientation did not change greatly. Fig. 8 shows the position in X, Y, Z and Fig. 9 shows the orientation in yaw, pitch and roll. The dotted lines are the measured data (ML) received from the left hand. The dash-dotted lines are the measured data (MR) received from the right hand. The star lines are the estimated data of the left robot end-effector (EL). The plus lines are the estimated data of the right robot end-effector (ER). The number of particles was set as 100. As a result, the KF could estimate the accurate orientation, and the PF could estimate the accurate position. The estimation algorithms could reduce the noise and improve the operating accuracy. Since the gap between the bolt and the nut was 0.5 mm, whether the operator could screw the bolt into the nut or not was used to evaluate the accuracy of the methods. Moreover, we compared the efficiency between our method and the methods in [11] and [12], including the operation time and times of faults in each test.

The three methods were compared in terms of operation time and times of faults. The operation time includes closing the two robot end-effectors, adjusting the two robot end-effectors, screwing the bolt, and separating the two robot end-effectors. The times of faults are the number of operators that tries to screw the bolt into the nut in each test.

Table III shows the results of the operation time and the times of faults for the five tests, respectively. For the test results of our method, the operation time for the five tests ranges from 236 to 245 s, with a mean time of 240.2 s. For the methods in [11] and [12], the mean times are 285.4 and 325.4 s, respectively. The mean times of faults are 3.6 and 5.6. Compared with the methods in [11] and [12], the mean time of our method drops about 45.2 and 85.2 s. Moreover, the times of faults of our method drops about 2.19 and 4.19.

VIII. DISCUSSION

This paper has proposed a robot control method based on the markerless tracking technology. The movements of robot manipulators are controlled by imitating the movements of the operator. The method shown here was testified on tracking a path and screwing a bolt into a nut. This system includes the operator into the decision control loop, which is a significant advantage. It allows a robot to finish some tasks such as moving without any prior knowledge of the start location and even destination location. Moreover, it is expected that this system can be used to achieve those complex postures when the joints of the robot are limited.

Compared with contacting electromagnetic devices, such as hand joystick and data glove, our method would not hinder most natural human-limb movements and would allow the operator concentrating on his own task instead of decomposing task into some simple commands. Compared with the noncontacting markerless method (such as that of Kofman *et al.* [11] and Du [12]), our method was proved to be more accurate and efficient. Moreover, as the manipulation task is more complex, multiple-robot cooperation will become a trend [29]. In addition, the

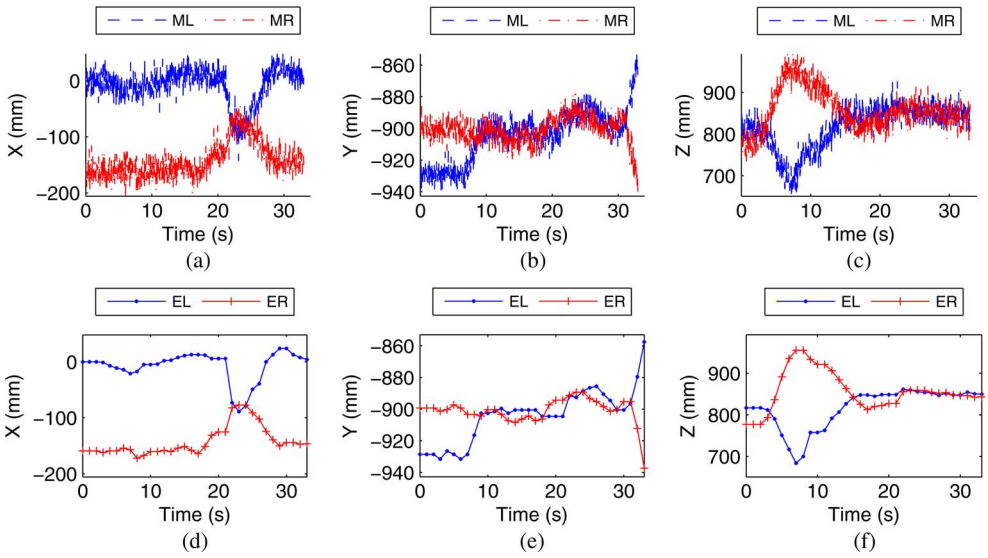

Fig. 8. Peg tracking results of Position measurements and estimations.

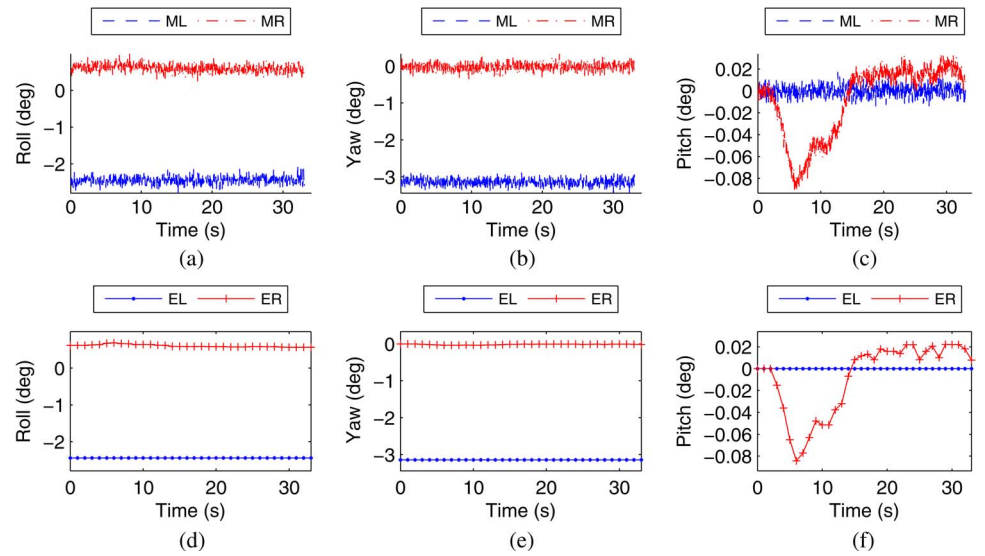

Fig. 9. Peg tracking results of Orientation measurements and estimations.

TABLE III
OPERATION TIME AND TIMES OF FAULTS OF THREE METHODS

		1	2	3	4	5	Mean
Our method	Time / s	245	241	236	239	240	240.2
	Times	2	2	1	1	1	1.41
Method [12]	Time / s	291	278	274	318	266	285.4
	Times	4	3	3	5	3	3.6
Method [11]	Time / s	331	298	294	368	336	325.4
	Times	6	5	5	7	5	5.6

contacting interfaces used in teleoperation for the multiple-robot manipulators often require multiple operators (as shown in [1]). In this method, it only needs one operator to control two robot manipulators to complete a complex manipulation. In the future, gesture control will be combined with voice control. Simple commands such as start, end, and pause can be controlled by voice. Then, the system will work more flexibly and advantageously.

IX. CONCLUSION

Teleoperation provides much probability to complete tasks that cannot be completely automated. This paper has focused on the need to provide a human–robot interface that uses a markerless tracking technology. Although it limits the movement space of human hand, this method provides convenience that result in better ergonomics, precision, and execution time. The human–robot interface can be applied to deep-sea robot, space robot and so on.

In this paper, KFs and PFs have been developed to estimate the orientation and position information of the human hands. The AMT method is employed to assist operators to complete high-precision tasks. In the experiment, screwing tasks have been carried out. We compared the efficiency of our method with the methods in [11] and [12]. Experimental results show that our method has better accuracy and efficiency.

Further investigation of force feedback would be considered to improve the human–robot interaction method.

REFERENCES

[1] K. B. Cho and B. H. Lee, "Intelligent lead: A novel HRI sensor for guide robots," *Sensors*, vol. 12, no. 6, pp. 8301–8318, Jun. 2012.

[2] T. Ando, R. Tsukahara, and M. Seki, "A hapic interface 'force blinker 2' for navigation of the visually impaired," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4112–4119, Oct. 2012.

[3] K. Kiguchi, S. Kariya, and K. Watanabe, "An exoskeletal robot for human elbow motion support-sensor fusion, adaptation, and control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 353–361, Aug. 2002.

[4] J. Kofman, X. Wu, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Trans. Ind. Electron.*, vol. 52, no. 5, pp. 1206–1219, Sep. 2005.

[5] V. Siddharth, "Vision-Based markerless 3-D Human-Arm tracking," M.S. thesis, Dept. Mech. Eng., Univ. Ottawa, Ottawa, ON, Canada, 2004, pp. 77–165.

[6] K. C. C. Peng, W. Singhose, and D. H. Frakes, "Hand-motion crane control using radio-frequency real-time location systems," *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 3, pp. 464–471, Feb. 2012.

[7] X. Suau, J. Ruiz-Hidalgo, and J. R. Casas, "Real-time head and hand tracking based on 2.5D data," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 575–585, Jul. 2012.

[8] M. Khezri and M. Jahed, "A neuro-fuzzy inference system for sEMG-based identification of hand motion commands," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1952–1960, Jun. 2011.

[9] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, "A hand-pose estimation for vision-based human interfaces," *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 676–684, Jul. 2003.

[10] G. Du, P. Zhang, and D. Li, "Human-manipulator interface based on multisensory process via Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5411–5418, Jan. 2014.

[11] J. Kofman, S. Verma, and X. H. Wu, "Robot-manipulator teleoperation by markerless vision-based hand-arm tracking," *Int. J. Optomechatron.*, vol. 1, no. 3, pp. 331–357, 2007.

[12] G. Du and P. Zhang, "Markerless human-robot interface for dual robot manipulators using kinect sensor," *Robot. Comput. Integr. Manuf.*, vol. 30, no. 2, pp. 150–159, Apr. 2014.

[13] S. P. Won, W. W. Melek, and F. Golnaraghi, "A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system," *IEEE Trans. Ind. Electron.*, vol. 57, no. 5, pp. 1787–1798, Sep. 2010.

[14] S. P. Won, W. W. Melek, and F. Golnaraghi, "A fastening tool tracking system using an IMU and a position sensor with Kalman filters and a fuzzy expert system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1782–1792, Nov. 2009.

[15] T. Khalid, Z. Mourad, C. Jean-Bernard, and B. Mohammed, "Bayesian bootstrap filter for integrated GPS and dead reckoning positioning," in *Proc. IEEE Int. Symp. Ind. Electron.*, Vigo, Spain, 2007, pp. 1520–1524.

[16] S. P. Won, W. Melek, and F. Golnaraghi, "Fastening tool tracking system using a Kalman filter and particle filter combination," *Meas. Sci. Technol.*, vol. 22, no. 12, pp. 1–12, Dec. 2011.

[17] Leap Motion, Aug. 2014. [Online]. Available: https://www.leapmotion.com/

[18] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, May 2013.

[19] X. Yun, E. R. Bachmann, and R. B. McGhee, "A simplified quaternion-based algorithm for orientation estimation from earth gravity and magnetic field measurements," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 3, pp. 638–650, Mar. 2008.

[20] M. H. Kim, S. Lee, and K. C. Lee, "Kalman predictive redundancy system for fault tolerance of safety-critical systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 46–53, May 2010.

[21] K. Y. Chan, C. K. F. Yiu, T. S. Dillon, and S. Nordholm, "Enhancement of speech recognitions for control automation using an intelligent particle swarm optimization," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 869–879, Mar. 2012.

[22] C. Yuan and M. J. Druzdzel, "Theoretical analysis and practical insights on importance sampling in Bayesian networks," *Int. J. Approx. Reason.*, vol. 46, no. 2, pp. 320–333, Oct. 2007.

[23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005, pp. 125–177.

[24] L. M. Munoz and A. Casals, "Improving the human-robot interface through adaptive multispace transformation," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 1208–1213, Jun. 2009.

[25] G. Antonelli, S. Chiaverini, and G. Fusco, "A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 162–167, Feb. 2003.

[26] R. Marin, P. J. Sanz, and R. Wirz, "A multimodal interface to control a robot arm via the web: A case study on remote programming," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1506–1521, Dec. 2005.

[27] J. Wang, "Stochastic modeling for real-time kinematic GPS/GLONASS position," *Navigation*, vol. 46, no. 4, pp. 297–305, 2000.

[28] W. Ding, J. Wang, and C. Rizos, "Improving adaptive Kalman estimation in GPS/INS integration," *J. Navigat.*, vol. 60, no. 3, pp. 517–529, Sep. 2007.

[29] D. F. Glas, T. Kanda, and N. Hagita, "Teleoperation of multiple social robots," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 530–544, Sep. 2012.

Guanglong Du received the Ph.D. degree from the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, in 2013.

He is currently a Postdoctoral Researcher with the School of Mechanical and Automotive Engineering, South China University of Technology. His research interests include machine vision, human–computer interaction, and virtual fixtures.

Ping Zhang received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degrees in robotics from Tianjin University, Tianjin, China, in 1985, 1988, and 1994, respectively.

He is currently a Professor with the Computer Science and Engineering School, South China University of Technology, Guangzhou, China. His research interests include intelligent networked robotics, intelligent networked manufacturing, human–computer interaction, and real-time embedded systems.