One New Human-Robot Cooperation Method Based on Kinect Sensor and Visual-Servoing

Hongbin Ma[1,2(✉)], Hao Wang[1], Mengyin Fu[1,2], and Chenguang Yang[3]

[1] School of Automation, Beijing Institute of Technology,
Beijing 100081, People's Republic of China
mathmhb@gmail.com
[2] State Key Lab of Intelligent Control and Decision of Complex Systems,
Beijing Institute of Technology, Beijing 100081, People's Republic of China
[3] School of Computing and Mathematics,
Plymouth University, Plymouth PL4 8AA, UK

Abstract. Human-robot interactions have received increased attention during the past decades for conveniently introducing robot into human daily life. In this paper, a novel Human-robot cooperation method is developed, which falls in between full human control and full robot autonomy. The human operator is in charge of the main operation and robot autonomy is gradually added to support the execution of the operator's intent. The proposed cooperation method allows the robot to accomplish tasks effectively with the help of human. The effectiveness of the method is verified by experiments, which is based on the Microsoft Kinect Sensor and the Virtual Robot Experimentation Platform.

1 Introduction

In recent years, the presence of robotic systems more or less evolved is common in our daily life. Since the start of this field of study from the last century, their evolution was incomparable. In the domains such as the automobile industry or the mobile phone factory, the uses of robots have become essential [1]. Humans and robots can perform tasks together and their relation became more important than a basic remote control to realize a task .The human/robot interaction is a large research area, which has attracted more and more interests in the research community.

The development of robots that incorporate teleoperation brings together two very different branches of human-robot interaction (HRI). One branch is social HRI, which focuses on studying psychological aspects of conversational interactions between people and robots. The human's role is as a service receiver in the interaction, in contrast to the robot's role as a service provider. The other branch is HRI for teleoperation, which typically focuses on issues like the workload of the operator (person remotely controlling the robot), situation awareness, and shared autonomy [2] for the remote operation of nonsocial robots.

This work is partially supported by National Natural Science Foundation (NSFC) under Grants 61473038 and 61473120. Also partially supported by Guangdong Provincial Natural Science Foundation of China 2014A030313266.

H. Liu et al. (Eds.): ICIRA 2015, Part I, LNAI 9244, pp. 523–534, 2015.
DOI: 10.1007/978-3-319-22879-2_48

Some researchers developed HRI systems using force sensor to control the motion of the robot. Mykoniatis developed a face recognizing robot to increase situation awareness and enhance HRI [3]. Wang designed a robot dancer control system to enhancing haptic HRI [4]. The system is based on force sensor feedbacks and laser range finders. However, the success rate is low and the controller needs further improvement. Mora developed a teleoperation approach for mobile social robots [5]. This approach incorporates automatic gaze control and three-dimensional spatial visualization. Wakita developed a motion control of an omnidirectional-type cane [6]. The system is based on a human-walking-intention model and force sensor. Spexard focused on achieve comprehending human-oriented interaction with an anthropomorphic robot [7]. This research brought together different interaction concepts and perception capabilities integrated on a humanoid robot. These systems cost lower than previous HRI systems, and enhanced the stability of the HRI, but limited in the success rate.

Among human-robot interaction techniques, visual interaction is dealt with as a major topic. In particular, Microsoft's Kinect sensor [7], which provides wealth of information like depth, which a normal video camera fails to provide. In addition, RGB-D data from the Kinect sensor can be used to generate a Skeleton model of humans with semantic matching of 15 body parts. Since human activities are a collection of how different body parts move across each time period, these information can be used to detect human activities.

The existing human-robot interaction systems enhance the stability and reduce the cost, but also have some limitations, such as comfort, accuracy and success rate. One solution to enhance the accuracy and success rate is to bridge the gap between full human control and full robot autonomy, the human operator is in charge of the main operation and robot autonomy is gradually added to support the execution of the operator's intent. Our objective is based on this idea. We aimed at enabling intuitive interaction between human and robot, in the context of an service scenario, where the two can collaborate to realize the task accurately. We will explore how a human-robot team can work together effectively.

The rest of this paper presents the cooperation system environment and details of the technique developed to accomplish the human-robot cooperation for the robotic application.

2 System Architecture

As an example of human-robot interaction, the objective of the human-robot cooperation system is to enable a robot to classify the objects of different shapes and colors on the table effectively with the direction of human. Note that in this experiment, the robot itself has no *priori* information of the position of the objects and the robot is not programmed in advance for achieving the task of classifying the objects. In other words, we aim to make the robot capable to achieve different unprogrammed jobs under the guidance of the operator, while the operator needs only to deliver his or her intent via rough motion intent. The required operations can be summarized as following:

– for the human: to direct the robot move arm to the nearby of the objects on the table.
– for the robot: to detect the objects and adjust the position of the gripper, then, grasp the objects and classify them.

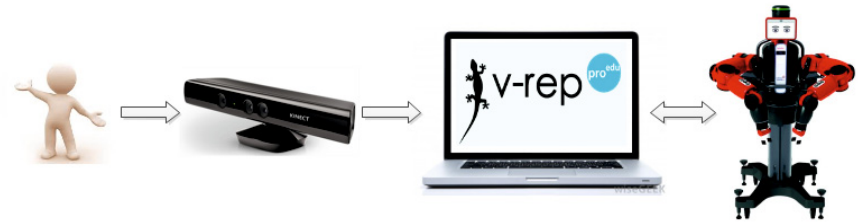

Fig. 1. The structure of the human-robot cooperation system, which includes Kinect Sensor and Baxter® robot and the Virtual Robot Experimentation Platform

The proposed cooperation system is composed of a Laptop, one Microsoft Kinect Sensor and the Virtual Robot Experiment Platform software. The Microsoft Kinect Sensor is used to detect and track human movement, these video data gathered by the Kinect with 640×480 pixels at 30 Hz are streamed to a laptop using a USB 2.0 connection. These data as the raw control signal are analyzed and then conversed into control command of each joint of the robot by the remote OpenNI API, sent to the V-REP over serial port communication.

The service robot shown in Fig. 1 is Baxter® robot in V-REP. The humanoid Baxter® robot includes a torso based on a movable pedestal and two 7DOF (degree of freedom) arms installed on left/right arm mounts respectively. Each arm has 7 rotational joints and 8 links, as well as an interchangeable gripper (such as electric gripper or vacuum cup) which can be installed at the end of the arm. A head-pan with a screen, located on the top of torso, can rotate in the horizontal plane.

The control object is the Baxter robot's arms in V-REP. The robot simulator V-REP, with integrated development environment, is based on a distributed control architecture. Next to offering the traditional approaches also found in other simulators, V-REP adds several additional approaches which makes V-REP very versatile and ideal for multi-robot applications.

3 Tele-Operation

3.1 Control Signal Acquisition

The control signal of the teleoperation system is from skeleton data tracked by the Kinect, the first feature can be extracted is joint data. For each joint, we have three main information. The first information is the index of the joints. Each joint has a unique index value. The second information is the positions of each joint in x, y, and z coordinates. These three coordinates are expressed in meters. The x, y, and z axes are the body axes of the depth sensor. This is a

right-handed coordinate system that places the sensor array at the origin point with the positive z axis extending in the direction in which the sensor array points. The positive y axis extends upward, and the positive x axis extends to the left (with respect to the sensor array).

3.2 Communication

After acquisition of the control signal, the next step is to send signal to V-REP. The communication between the OpenNI and the Virtual Robot Experimentation Platform has to be real time. V-REP allows the user to choose among various programming techniques simultaneously and even symbiotically. The cooperation system chooses the remote API client method to make V-REP to communicate with OpenNI.

The remote OpenNI API plugin start as the 'Kinect Server' communicating via a free port set in List. 1 before. The temporary remote API server service was chosen to start the 'Kinect Server' from within a script at simulation start. After the simulation started, the child script will detect the free serial port set in the 'Kinect Server' and receive skeleton data from the 'Kinect Server'.

3.3 Robot Arm Tele-Control

To remotely control the robot, the first required development was the arm control. Offsets were applied to define the workspace of the virtual arm. They provided a suitable workspace environment where the remote control of the Baxter® Robot arm was possible. Then the choice concerning was decided. As a mirror, the arms movements were matched to the Baxter® Robot Simulator arms, as the user moved its own arms. The Cartesian systems used for the Baxter® Robot Simulator and the Human in Real World are shown in Fig. 2.

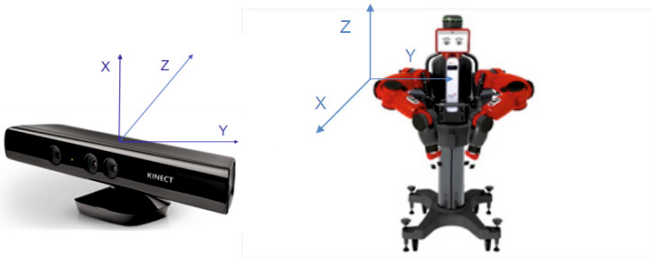

Fig. 2. The Cartesian world of Kinect Sensor and simulator

The control of the arms was performed by sending positions of shoulders, elbows and hands with a command via a free serial port to the Baxter® Robot Simulator. Then it moved its joints according to its inverse kinematics in order to reach the desired positions. In order to get the target positions in simulator, the coordinate transformation is performed from the real world space coordinates to the task space coordinates.

More specifically, the OpenNI tracker detects the position of the following set of joints in the 3D space $G = \{g_{i,\ i\in[1,I]}\}$. The position of joint g_i is implied by vector $P_{i0}(t) = [x_m\ y_m\ z_m]^T$, where t denotes the frame for which the joint position is located and the origin of the orthogonal X, Y and Z coordinates system is placed at the center of the Kinect sensor. The task space is also a 3-dimensional space, the position of joint g_i is implied by vector $P_{i1}(t) = [x_s\ y_s\ z_s]^T$ in the task space. The work space mapping is shown as below:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \left(\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} + \begin{bmatrix} T_x \\ T_x \\ T_z \end{bmatrix} \right) \tag{1}$$

where δ is the revolution angle about the Y-axis of the manipulator base frame, $[S_x\ S_y\ S_z]^T$ and $[T_x\ T_y\ T_z]^T$ are the scaling factors and translations about the X, Y and Z axis. The mapping parameters of (1) are given by

$$\delta = \frac{\pi}{2}, \quad \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.7 \end{bmatrix}, \quad \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} 0.051 \\ 0.049 \\ 0.034 \end{bmatrix}. \tag{2}$$

4 Visual Servoing System

In order to implement the function that the Baxter® robot can classify the objects of different shapes and colors on the table effectively while receives the trigger signal, the image-based visual servoing system [8] was applied. The visual servoing system consists of one fixed camera and two cameras on each wrist of the Baxter® robot that can image a scene containing the different objects on the table.

4.1 Image Jacobian

The image Jacobian matrix relates robot joint angle changes (or robots positions) to image feature changes. For an eye-in-hand system, if the object in the workspace is motionless, the image feature changes would be only related to the motion of the robot or camera, which can be expressed as

$$y = f(\theta) \tag{3}$$

Suppose $\theta = [\theta_1, \theta_2, \cdots, \theta_n]^T$denotes the coordinate in robot joint space, $\mathbf{s} = [s_1, s_2, \cdots, s_p]^T$ is that of the robot end-effector in Cartesian space, and $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$ is image feature vector. Consequently $\dot{\mathbf{s}}$ is the velocity of robot end-effector including translation and rotation, $\dot{\theta}$ is the velocity in robot joint space, and $\dot{\mathbf{y}}$ is the velocity in image feature space. The relation between $\dot{\theta}$ and $\dot{\mathbf{s}}$ is

$$\dot{\mathbf{s}} \approx \mathbf{J}_\alpha \cdot \dot{\theta} \tag{4}$$

where

$$\mathbf{J}_\alpha = \frac{\partial \mathbf{s}}{\partial \theta} = \begin{bmatrix} \frac{\partial s_1}{\partial \theta_1} & \frac{\partial s_1}{\partial \theta_2} & \cdots & \frac{\partial s_1}{\partial \theta_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial s_p}{\partial \theta_1} & \frac{\partial s_p}{\partial \theta_2} & \cdots & \frac{\partial s_p}{\partial \theta_n} \end{bmatrix} \tag{5}$$

which is called robot Jacobian or feature Jacobian [9].

Similarly, the relation between $\dot{\mathbf{s}}$ and $\dot{\mathbf{y}}$ can be

$$\dot{\mathbf{y}} \approx \mathbf{J}_\beta \cdot \dot{\mathbf{s}} \tag{6}$$

where

$$\mathbf{J}_\beta = \frac{\partial \mathbf{s}}{\partial \theta} = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_1} & \frac{\partial y_1}{\partial \theta_2} & \cdots & \frac{\partial y_1}{\partial \theta_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial y_p}{\partial \theta_1} & \frac{\partial y_p}{\partial \theta_2} & \cdots & \frac{\partial y_m}{\partial \theta_n} \end{bmatrix} \tag{7}$$

J_β is called image Jacobian [10], interaction matrix [11], or feature Jacobian. Thus we can get $\dot{\mathbf{y}} = \mathbf{J}_\beta \cdot \dot{\mathbf{s}} = \mathbf{J}_\beta \cdot \mathbf{J}_\alpha \dot{\theta}$. Let $\mathbf{J}_\theta = \mathbf{J}_\beta \cdot \mathbf{J}_\alpha$, then

$$\dot{\mathbf{y}} \approx \mathbf{J}_\theta \cdot \dot{\theta} \tag{8}$$

$\mathbf{J}_\theta$ is nominated as composite Jacobian or Visual-Motor Jacobian.

4.2 Online Estimation with Kalman Filtering

In an image-based robot visual servo system, the Jacobian matrix is a complex temporal nonlinear matrix. Its accuracy would affect directly to the entire system performance. Online estimation of of Jacobian matrix is to acquire the Jacobian matrix by dynamic computing based on real-time measurements of the robot jointangulars and observations of the image features [12]. The system model is not required in advance.

By a constructing state space model, the Jacobian estimation problem is transformed to that of system state estimation. The system state vector is composed of elements of Jacobian matrix. Furthermore, image features construct the measurement vector of the system. Kalman filtering algorithm applied to estimate the states. In order to estimate elements of total Jacobian matrix, measurement vector x is defined as $mn \times 1$vector, which is as follows

$$\mathbf{x} = [\mathbf{J_1}\ \mathbf{J_2}\ \cdots\ \mathbf{J_m}]^T \tag{9}$$

where $\mathbf{J_i} = [\frac{\partial y_i}{\partial \theta_1} \frac{\partial y_i}{\partial \theta_2} \cdots \frac{\partial y_i}{\partial \theta_n}]$is the i^{th} row vector of Jacobian matrix $\mathbf{J}$.

Let $x(k)$ as system state, the difference of image feature as system's output, that is

$$z(k) = y(k+1) - y(k) \tag{10}$$

According to equation (10), we could get the state space model

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{x}(k) + \omega(k) \\ z(k) = \mathbf{H}(k)x(k) + \upsilon(k) \end{cases} \tag{11}$$

where

$$\mathbf{H}(k) = \begin{bmatrix} \Delta\mathbf{h}(k)^T & & 0 \\ & \ddots & \\ 0 & & \Delta\mathbf{h}(k)^T \end{bmatrix}_{m \times mn} \tag{12}$$

$\omega(k)$ and $\upsilon(k)$ are process noise and measurement noise respectively.

If the distribution of system noises is Gaussian, the following recursive estimation can be obtained by using Kalman filtering algorithm:

$$\begin{cases} \hat{\mathbf{x}}^-(k+1) = \hat{\mathbf{x}}(k) \\ \mathbf{P}^-(k+1) = \mathbf{P}(k) + \mathbf{R}_\omega \\ \mathbf{K}(k+1) = \mathbf{P}^-(k+1)\mathbf{H}^T(k)[\mathbf{H}(k)\mathbf{P}^-(k+1)\mathbf{H}^T(k) + \mathbf{R}_\upsilon]^{-1} \\ \mathbf{P}(k+1) = [I - \mathbf{K}(k+1)\mathbf{H}(k)]\mathbf{P}^-(k+1) \\ \hat{\mathbf{x}}(k+1) = \hat{\mathbf{x}}^-(k+1) + \mathbf{K}(k+1)[\mathbf{z}(k+1) - \mathbf{H}^T(k)\hat{\mathbf{x}}(k)] \end{cases} \tag{13}$$

R_ω,R_υ are noise covariance matrixes, $\mathbf{P}(k)$ is state estimation error covariance matrix, whose initial value can be $\mathbf{P}(0) = 10^5\mathbf{I}_{mn}$.

The calculation process of Kalman filtering algorithm could be divided into two steps: time update and measurement update. Time update process is responsible for proceeding with the calculation of present state variable and estimating error covariance, in order to provide prior estimation for the next state. Measurement update is responsible for feedback, that is, it combines prior estimation and new measurement to construct updated posterior estimation. Time update equation is also called pre-estimation equation; measurement update equation is also called revised equation.

4.3 Control Algorithm

The task of visual servo is to control the motion of a robot via proper control law according to the measured image feature error, and eventually make the image feature error approximate to zero. The definition of image feature error is:

$$\mathbf{e} = \mathbf{y}_d - \mathbf{y} \tag{14}$$

The control is designed as:

$$\dot{\theta}_c = \hat{\mathbf{J}}_\theta^+\mathbf{e} - \hat{\mathbf{J}}_\theta^+\hat{\mathbf{J}}_t, \quad \hat{\mathbf{J}}_\theta^+ = (\hat{\mathbf{J}}_\theta^T\hat{\mathbf{J}}_\theta)^{-1}\hat{\mathbf{J}}_\theta^T \tag{15}$$

The second term of the control law formula (15) is to compensate for the image feature changes induced by the object motion. If the object is motionless, the

second term will be zero and the visual control task will become object positioning with visual feedback. If the object is moving, the visual control task will be object tracking with visual feedback.

Suppose the sample interval is Δt , then the discrete control law is:

$$\Delta\theta(k) = \lambda\hat{\mathbf{J}}_{\theta}^{+}(k)\mathbf{e}(k) - \hat{\mathbf{J}}_{\theta}^{+}(k)\hat{\mathbf{J}}_{t}(k)\Delta t, \quad \lambda > 0 \tag{16}$$

The current image feature error is obtained through the contrast between the current image feature and the desired image feature. The changes of image feature errors and the robot joints are input into the corresponded filter and the current Jacobian matrix is calculated. At last, the robot is controlled by the control law according to Jacobian matrix and the image feature errors. The above processes repeated until the image feature error approximates to zero.

4.4 Inverse Kinematics

For a serial manipulator for instance, the inverse kinematics problem would be to find the value of all joints in the manipulator given the position (and/or orientation) of the end effector. The inverse kinematics problem can be simplified to solve the equation

$$\boldsymbol{e} = J\triangle\theta \tag{17}$$

where $\boldsymbol{e} = \boldsymbol{t} - \boldsymbol{s}$, the vector $\boldsymbol{t} = (t_1, \ldots, t_k)^T$ are the target positions, the vector $\boldsymbol{s}$ are the end effectors positions, and $\theta = (\theta_1, \ldots, \theta_n)^T$ are the joint angles.

There are many methods of modeling and solving inverse kinematics problems [13]. The inverse kinematics with end-effector posture constraints was proposed in order to implement the grasping motions accurately. There are kinds of methods for the requirements of end-effector constraints. We proposed one new inverse kinematics method combined the damped least squares method and the geometric method. First step of the new method is to calculate using the damped least squares method, the second step is to implement the end-effector posture constraints using geometric method.

The damped least squares method avoids many of the pseudo-inverse method's problems with singularities and can give a numerically stable method of selecting $\triangle\theta$. It is also called the Levenberg-Marquardt method and was first used for inverse kinematics by Wampler [14] and Nakamura and Hanafusa [15]. Rather than just finding the minimum vector $\triangle\theta$ that gives a best solution to equation (5), we find the value of $\triangle\theta$ that minimizes the quantity

$$\| J\triangle\theta - \boldsymbol{e} \|^2 + \lambda^2 \| \triangle\theta \|^2 \tag{18}$$

where $\lambda \in \mathbb{R}$ is a non-zero damping constant. This is equivalent to minimizing the quantity

$$\| \begin{pmatrix} J \\ \lambda I \end{pmatrix} \triangle\theta - \begin{pmatrix} \boldsymbol{e} \\ 0 \end{pmatrix} \| \tag{19}$$

The corresponding normal equation is

$$\begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} J \\ \lambda I \end{pmatrix} \triangle\theta = \begin{pmatrix} J \\ \lambda I \end{pmatrix}^T \begin{pmatrix} \boldsymbol{e} \\ 0 \end{pmatrix} \tag{20}$$

This can be equivalently rewritten as

$$(J^T J + \lambda^2 I)\triangle\theta = J^T \boldsymbol{e} \tag{21}$$

It can be shown that $J^T J + \lambda^2 I$ is non-singular. Thus, the damped least squares solution is equal to

$$\triangle\theta = (J^T J + \lambda^2 I)^{-1} J^T \boldsymbol{e} \tag{22}$$

Now $J^T J$ is a $n \times n$ matrix, where n is the number of degrees of freedom. It is easy to show that $(J^T J + \lambda^2 I)^{-1} J^T = J^T (JJ^T + \lambda^2 I)^{-1}$. Thus,

$$\triangle\theta = J^T (JJ^T + \lambda^2 I)^{-1} \boldsymbol{e} \tag{23}$$

The advantage of equation (23) over (22) is that the matrix being inverted is only $m \times m$ where $m = 3k$ is the dimension of the space of target positions, and m is often much less than n.

Additionally, (23) can be computed without needing to carry out the matrix inversion, instead row operations can find $\boldsymbol{f}$ such that $(JJ^T + \lambda^2 I)\boldsymbol{f} = (e)$ and then $J^T \boldsymbol{f}$ is the solution.

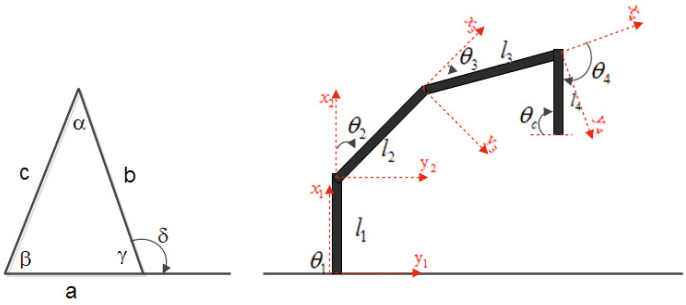

Fig. 3. The left is an arbitrary triangle and the right is the geometric based inverse kinematics

The geometric inverse kinematics method was applied to implement the end-effector posture constraints. The geometric method is based on exterior angle theorem and cosine rule. Given a triangle in the left of the Fig. 3, the exterior angle theorem says as in equations (24),a cosine rule says as in equations (25).

$$\delta = \alpha + \beta \tag{24}$$

$$b^2 = a^2 + c^2 - 2ac\cos\beta \tag{25}$$

Based on triangle in the left of the Fig. 3, an inverse kinematics based on geometric calculation is proposed. The right of the Fig. 3 redefine servomotor

angle in the arm robot frame. Inverse kinematic is then calculated by solving the angle in robot configuration with end-effector posture constraints. The angle θ_c represents the angle between the end-effector and the horizontal line. In our experiment scene, θ_c is set to $\frac{\pi}{2}$. The following equation is got to implement the end-effector posture constraints,

$$\theta_4 = \frac{\pi}{2} - \theta_2 + \theta_3 \tag{26}$$

5 Experiment and Results

Our goal is to explore whether a human-robot team can accomplish the service task together effectively using the proposed cooperation method. Five different objects were used to test the detecting and classifying accuracy of the Baxter® visual servoing system. The first part of the experiment is to examine the validity of the control of the robot arm with motion. We design 4 motions to observe the motion of subject and 3D robot simulator. The first picture of the Fig. 4 show the procedure of the robots bending its elbow joint. The second picture of the Fig. 4 show the control of the robot simulator whose arms swing up and down. In each subfigure of Fig. 4, the left side shows the motion of the subject detected by the Kinect based on the skeleton tracking technology, while the right side displays the motion of the simulated robot Baxter®From the figures, we find that the simulated robot Baxter® can track the motion of subject's arm effectively.

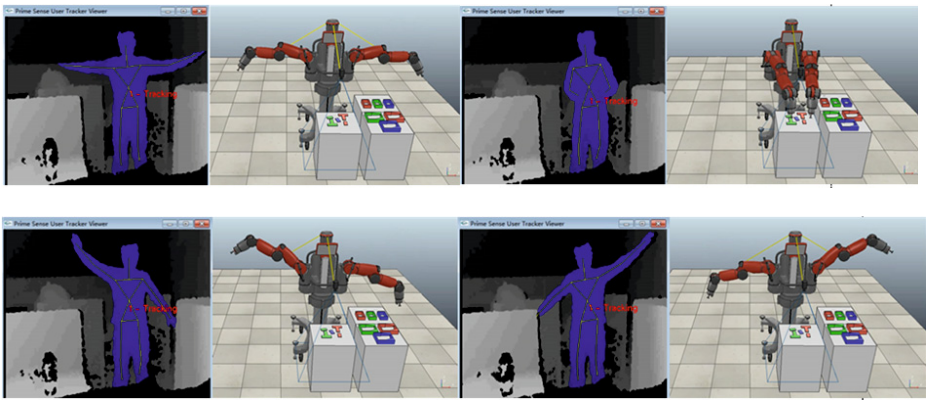

Fig. 4. The experiment results of teleopration simulation

One object tracking experiment is applied to test the result of the visual servoing system. The camera pinhole model is used and the image resolution is 480×480 pixels. The camera parameters are as follows. The focal length $f = 8mm$, the principal point is $(240, 240)$, and the scale factor in image horizontal axis and vertical axis are all $0.05mm/pixels$. Suppose the object is point-shaped, the selected image features are x-coordinate and y-coordinate in the image plane

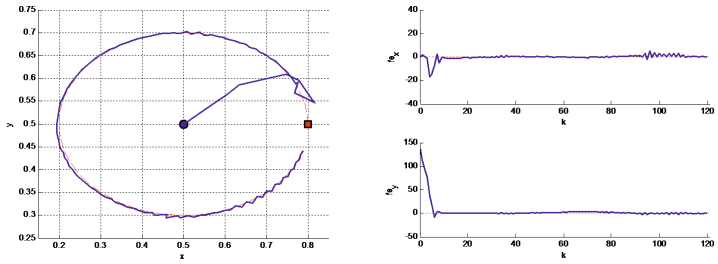

Fig. 5. The left is the robot tracking trajectory based on Kalman filter. The circle is the initial position of robot end-effector; The square is the initial position of object. Dashed: object's real motion trajectory; Crude real line: robot end-effector's motion trajectory. The right is the image feature error based on Kalman filter

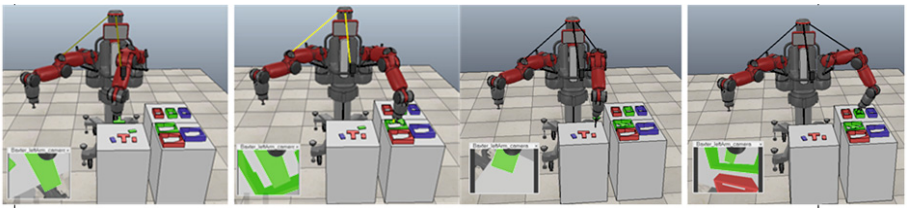

Fig. 6. The experiment results of the Baxter®: classifying function, Baxter® picks up them and classify them.

of the object. The controlling goal is to keep the robot end-effectors motion following the objects motion.

In order to verify the accuracy of the blob detection module, we put the objects on different positions. Fig. 6 illustrates that the Baxter® robot can pick up objects and classify them effectively.

6 Conclusion

In this paper, we have put forward one novel Human-robot cooperation method which falls in between full human control and full robot autonomy. In order to verify the effectiveness of the method by experiments, an experiment environment based on the Kinect sensor and V-REP was built up. In the experiments, the Kinect sensor was used to obtain the skeleton framework information of the human, then, the data is classified and transformed to the robot as the motion control signal. Moreover, the Blob detector was applied to detect the target and applied the closed loop control to implement autonomous adjustment in the visual servoing system. It can be found that the human-robot team can work effectively through the experiments.

References

1. Renon, P., Yang, C., Ma, H., Cui, R.: Haptic interaction between human and virtual icub robot using novint falcon with chai3d and matlab. In: 2013 32nd Chinese Control Conference (CCC), pp. 6045–6050. IEEE (2013)
2. Pitzer, B., Styer, M., Bersch, C., DuHadway, C., Becker, J.: Towards perceptual shared autonomy for robotic mobile manipulation. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 6245–6251. IEEE (2011)
3. Sun, D., Wang, C., Shang, W., Feng, G.: A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. IEEE Transactions on Robotics **25**(5), 1074–1086 (2009)
4. Wang, H., Kosuge, K.: Control of a robot dancer for enhancing haptic human-robot interaction in waltz. IEEE Transactions on Haptics **5**(3), 264–273 (2012)
5. Mora, A., Glas, D.F., Kanda, T., Hagita, N.: A teleoperation approach for mobile social robots incorporating automatic gaze control and three-dimensional spatial visualization. IEEE Transactions on Systems, Man, and Cybernetics: Systems **43**(3), 630–642 (2013)
6. Wakita, K., Huang, J., Di, P., Sekiyama, K., Fukuda, T.: Human-walking-intention-based motion control of an omnidirectional-type cane robot. IEEE/ASME Transactions on Mechatronics **18**(1), 285–296 (2013)
7. Spexard, T.P., Hanheide, M., Sagerer, G.: Human-oriented interaction with an anthropomorphic robot. IEEE Transactions on Robotics **23**(5), 852–862 (2007)
8. Hojaij, A., Zelek, J., Asmar, D.: A two phase rgb-d visual servoing controller. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), pp. 785–790. IEEE (2014)
9. Jang, W., Bien, Z.: Feature-based visual servoing of an eye-in-hand robot with improved tracking performance. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, pp. 2254–2260. IEEE (1991)
10. Asada, M., Tanaka, T., Hosoda, K.: Adaptive binocular visual servoing for independently moving target tracking. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2000, vol. 3, pp. 2076–2081. IEEE (2000)
11. Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation **8**(3), 313–326 (1992)
12. Lv, X., Huang, X.: Fuzzy adaptive Kalman filtering based estimation of image Jacobian for uncalibrated visual servoing. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2167–2172. IEEE (2006)
13. Sun, Z., He, D., Zhang, W.: A systematic approach to inverse kinematics of hybrid actuation robots. In: 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 300–305. IEEE (2012)
14. Wampler, C.W.: Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. IEEE Transactions on Systems, Man and Cybernetics **16**(1), 93–101 (1986)
15. Nakamura, Y., Hanafusa, H.: Inverse kinematic solutions with singularity robustness for robot manipulator control. Journal of Dynamic Systems, Measurement, and Control **108**(3), 163–171 (1986)