Robot teleoperation system based on SVDD

Bao Xi, Naijun Liu, Tao Lu, Jinyan Lu, and Yinghao Cai, Shuo Wang
The State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Haidian, Beijing 100190
Email: $\{xibao2016, liunaijun2016, shuo.wang, tao.lu, yinghao.cai\}$@ia.ac.cn

***Abstract*—This paper presents a real-time vision based robot teleoperation system that consists of a three-dimensional (3D) vision subsystem and a slave robot which are connected by LAN. The vision subsystem utilizes an Asus Xtion Pro Live camera to get the 3D data of the operation scene. The vision system is used to determine the position and orientation of a four-ball feature frame held by the operator. Then the position and the orientation are used to control a remote robot. In the vision subsystem, support vector domain description (SVDD) is adopted to detect the balls on the feature frame. In this paper, we propose a novel colour table to speed up the detection procedure and utilize Kalman filter for ball tracking to reduce the detection area for further acceleration. The operator can see the motion of the robot which enables the operator make some corrections constantly. The SVDD colour classifier and the teleoperation system are tested in experiments.**

***Index Terms*—SVDD, Kalman filter, Teleoperation, Position and orientation**

I. Introduction

With the development of computer vision and machine learning theory, robot is becoming more and more intelligent in recent years. However, there are still lots of tasks that robots cannot accomplish by itself independently, especially when dealing with complex and dangerous tasks, including handling nuclear facilities, dismantling bombs, underwater and space exploration, rescue applications in dangerous environments, and minimally invasive surgery and so on. Under these circumstances, remote human teleoperation is required[12].

In many occasions, operator uses control plane, dials, and computer graphical interfaces to manipulate a robot remotely, but these methods are usually unnatural for operator and need continuously practice to complete a teleoperation task. A more natural idea is mapping the motion of the operator to robot directly, such as mapping the operator hand-arm motion to a robot arm to accomplish a required task. In this procedure, a variety of sensors, such as exoskeletal mechanical devices[5], data glove with angle sensors[8][7][14], inertial motion tracking sensors[2][11] are used to capture the motion information of operator. Since the operator has to wear some devices, these methods are called contacting methods. However, a drawback of these contacting methods is that the sensors worn by an operator may hinder the motion of operator. By comparison, non-contact vision based techniques do not need the operator to wear extra devices, so they are less invasive and less hindrance to human motions. Marker free vision based methods just use one or more cameras to capture the motion of operator and convert the motion information to control information of remote robots. Many marker free methods, based on monocular vision system, can only provide the 2D motion information of the operator[10][13]. However, a remote robot usually requires 3D position and orientation information with respect to a specific reference system. There is also a variety of marker free methods control the robot by a number of simple hand gestures, such as left, right, up, down. Due to the limited number and the simple semantic character of hand gestures, it is difficult to control the remote robot to complete a sophisticated manipulation task. In order to make the operator to control the robot by hand-arm motion more naturally, Kofman et al. developed a method that control a robot manipulator using the three-dimensional human hand motion of the operator in real time[12]. However, this method requires a specific initialization posture with an unclothed arm in front of a black background, and there exist some time delay due to the image processing and robot motion execution. Guanglong Du at al. proposed a similar method but the hand motion of the operator was confined to a relatively small space[6], which leaded to losing some precision.

By comparison, the marker-based methods have the advantage of easy detecting and high precision. Mayez et al. used a feature frame, consisted of four balls with distinct colours, to get the positions and the orientations of the hand of operator[1]. In this method, operator has to initialize the reference RGB parameters for each colour by selecting the corresponding pixels manually in the continuously refreshed image at every run, which is both trivial and troublesome. In addition, in order to detect the colour ball, this method measures the similarity between the RGB components of the target pixel and the reference RGB components. It is an oversimplified and crude way since the RGB components of a given colour vary widely under different environment illumination. In this paper, an Asus Xtion Pro Live camera is used to monitor a feature frame that is held by the operator. The frame is similar with that in [1]. In the scenario of single colour classification, collecting all possible background colours is a redundancy work since we are only interested in four kind of prescribed colours. Therefore, one-class classifiers are more effective in this situation as they just need the training set of the prescribed colours. Support vector domain description (SVDD)[15][4] developed by Tax and Duin is an efficient one-class classifier, which is trained to map the training data into a high dimensional feature space to find a hyper-sphere to

978-1-5386-3524-7/17/$31.00 ©2017 IEEE

Fig. 1. Universal Robot 5 and a gripper

Fig. 2. Asus Xtion Pro Live Camera

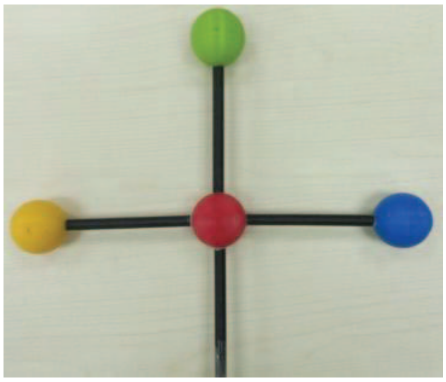

Fig. 3. Four-ball frame

enclose the training data as much as possible while keeping the volume of the hyper-sphere as small as possible. Thus, we use SVDD to detect the colour balls and propose a novel colour table to speed up the detection procedure. In order to reduce computation, instead of searching the whole image, we use Kalman filter[3] to predict the positions of the colour balls, and just search the corresponding regions based on the predictions.

The reminder of this paper is organized as follows. Section II introduces the hardware platform. In section III introduces a method based on SVDD to evaluate the positions and the orientations of the hand of operator and proposes a novel colour table to improve processing speed. Section IV gives two experiments about the SVDD colour classifier and the entire teleoperation system. Finally, section V makes a summary of this paper.

II. Hardware platform

The robot used in this work is the Universal Robot 5 (UR5) equipped with a two-finger adaptive gripper from Robotiq company. UR5 is a well-known 6-degree-of-freedom (DoF) robotic manipulator from Universal Robot company. It can be controlled by torque and position mode, and all its joints contribute to the transformational and rotational movements of its end effector[9]. In this work, UR5 is connected to the controller using TCP/IP socket.

The Asus Xtion Pro Live camera used in this proposed teleoperation system is comprised of a RGB camera and a dual infrared depth sensor. Using image and depth sensors, Xtion pro can capture the position orientation of the feature frame. Fig. 2 shows the Asus Xtion Pro Live camera.

In this work, a four-ball feature frame with different colours as shown in Fig. 3 is used to get the position and orientation of the hand of the operator. Since Xtion Pro is equipped with a depth sensor, we can get the 3-D positions of the balls of the frame in the scene. Then we can obtain the orientation of the frame.

The overall robot teleoperation system proposed in this paper consists of a master station and a slave robot station, which are connected through a LAN as is shown in Fig. 4. The vision system of the master station captures the hand motion of the operator and sends to UR5, and the vision system of the slave system provides the feedback of the operation of UR5.

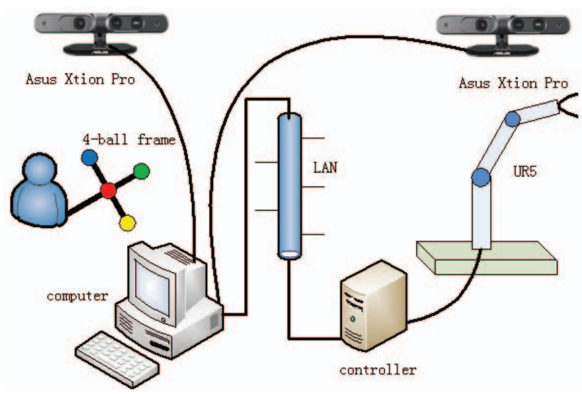

Fig. 4. Vision based robot teleoperation system

III. Software architecture

A. Support Vector Domain Description

Support vector domain description (SVDD) is a method for one-class classification, which defines a minimum hypersphere that characterized by a center and radius to enclose the samples in the training set as much as possible, and the hypersphere acts as the classification boundary.

Suppose the training set containing l samples is described as $X = \{x_1, x_2, \cdots, x_l\}$, where $x_i \in R^n$. The problem to obtain the minimum hypersphere defined by the SVDD is formulated as the following optimization problem:

$$\begin{aligned} \min_{a,R,\xi} \quad & R^2 + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & \|x_i - c\|^2 \le R^2 + \xi_i \\ & \xi_i \ge 0, \quad i = 1, 2, \cdots, l \end{aligned} \tag{1}$$

where R and c are the radius and the center of the hypersphere respectively, the variable ξ_i is a slack variable that controls the sensitivity of SVDD to the possible outliers, and the parameter C is a penalty factor that controls the trade-off between minimizing the volume of the hypersphere and satisfying the constraints for each sample.

The corresponding Wolf dual of formula (1) with Lagrange multipliers $\alpha_i \ge 0$ is[15]:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{l}\alpha_i \langle x_i, x_i\rangle - \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j \langle x_i, x_j\rangle \\ s.t. \quad & \sum_{i=1}^{l}\alpha_i = 1 \\ & 0 \le \alpha_i \le C, \quad i = 1, 2, \cdots, l \end{aligned} \tag{2}$$

The center of the hypersphere c is a linear combination of the samples in the training set weighted by the Lagrange multipliers α_i:

$$c = \sum_{i=1}^{l}\alpha_i x_i \tag{3}$$

The Lagrange multipliers α_i are obtained by solving (2). However, only a small fraction of α_i is non-zero, and the corresponding x_i is called support vector, which locates on the hypersphere. The radius R of the hypersphere is obtained by calculating the distance from the center of the hypersphere to one of the support vectors with the corresponding $0 \le \alpha_i \le C$.

With the radius R and centre c, we can determine whether a test sample is inside the hypersphere by calculating the distance to the centre c and comparing this distance with the radius R. The test sample is accepted when the distance is shorter than R.

The square of the distance to the centre is calculated by inner product:

$$f(x) = \langle x - c, x - c\rangle \tag{4}$$

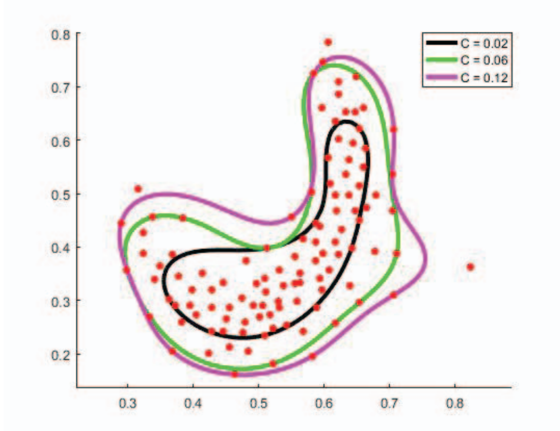

Fig. 5. Classification boundaries of SVDD with different C

The inner product in (4) can be replaced by a Mercer kernel $\kappa(\bullet, \bullet)$. In this paper, a Gaussian kernel with kernel width σ is adopted, and then the inner product is expressed as:

$$\kappa(x, u) = exp\left(-(x-u)^2/2\sigma^2\right) \tag{5}$$

B. A Novel Colour Table

Since the feature frame is consisted of four balls with distinct colours, we have to train four different SVDD classifiers respectively. In order to detect the balls, the four SVDD classifiers classify every pixel. Therefore, the computation burden is extremely heavy and cannot meet the real-time requirement. In this paper, we come up with a novel colour table to reduce the computation burden.

Compared to RGB colour space, YCbCr coding is more robust to the variation of ambient illumination. Therefore, detecting the balls in YCbCr colour space will have an advantage. In a 24-bit colour image, each pixel is represented by a triplet, such as (Y, Cr, Cb) or (R, G, B), which supports 256×256×256 (16,777,216) possible combined colours. In this scenario, the colour in the YCbCr colour space is enumerable. Therefore, we can use the acquired four SVDD classifiers to classify all the 16,777,216 colours in YCbCr space in advance and save the classification results to a colour table. As a result, we can classify a pixel in an image by just looking up the colour table. Since looking up the colour table takes only an extremely short time, the procedure to detecting the colour balls can meet the real-time requirement.

The structure of the proposed colour table, consisted of 3 channels, is as shown in Fig. 6. The size of the colour table is 256×256×3. In order to distinguish between colours, we number the four different colours, namely blue, green, yellow, and red, from 1 to 4, and number all the remaining colours 0. The column and row of the colour table represent the Cr and Cb component respectively, and channel 2 and channel 3 of the colour table give the lower bound and upper bound of Y component for specific Cr and Cb pairs. When determining the colour of a pixel (Y, Cr, Cb), we first look up channel 1 to get the integer in position (Cr, Cb), and then examine whether

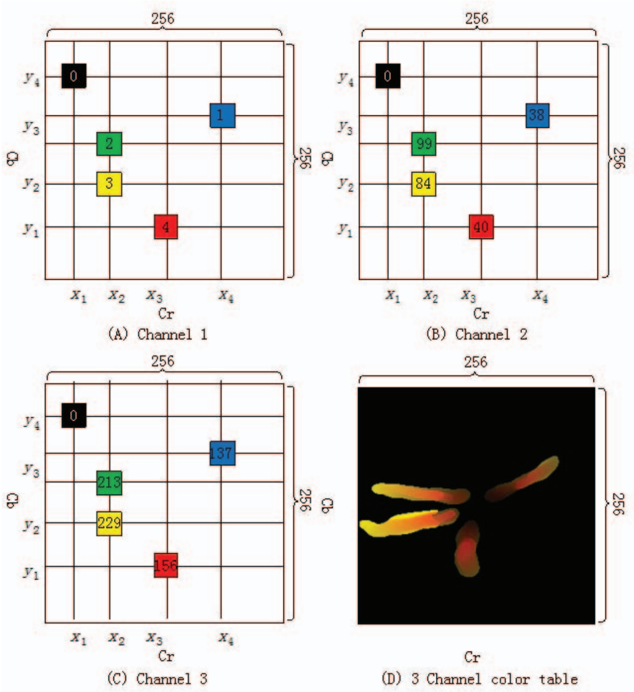

Fig. 6. Classification boundaries of SVDD with different C

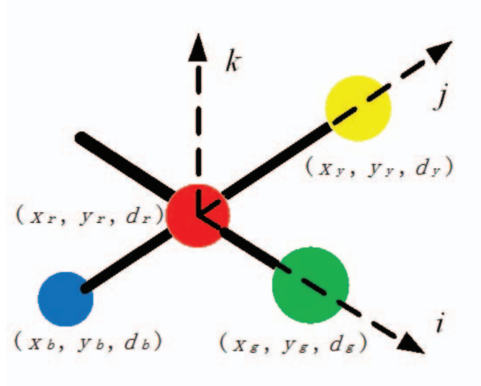

Fig. 7. The coordinate system based on the frame

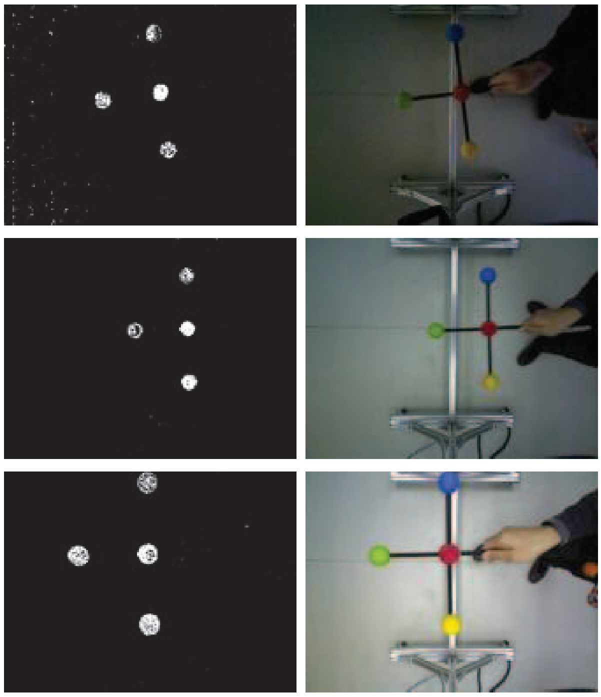

Fig. 8. The detection of balls in different illumination

Y is between the corresponding lower bound and upper bound in position (Cr, Cb) by looking up channel 2 and channel 3. For example, if $Cr = x_1$ and $Cb = y_4$, we know that it is background colour from the channel 1 of the colour table. When $Cr = x_2$ and $Cb = y_2$, the corresponding value in channel 1 is 3, so it may be a yellow colour. And then we check channel 2 and channel 3, if $84 \leq Y \leq 229$, it is a yellow colour. Therefore, compared to classify the colour of a pixel by SVDD classifier directly, look-up table method based on this 3-channel colour table is more efficient and time saving. After classifying all pixels in an image, the positions of each balls are determined.

In order to reduce the computational burden further, a Kalman filter is adopted to predict the position of each ball in the next frame so that the detection can be done in specific small regions based on the predictions.

C. *Get the Position and Orientation*

As the Asus Xtion Pro live provides both colour image and depth image, we can read the depth of each ball from the depth image after getting their positions in the colour image. Suppose the 3D positions of each ball are as shown in Fig. 7. To get the pose of the frame, a rectangular coordinate system is established on the frame. Since the positions of the balls have been obtained, the base vectors of the coordinate system can be easily calculated. Based on these three base vectors, we can obtain the pose of the frame. Then we use this pose and the position of the red ball to approximate the position and the orientation of the hand that hold the frame.

IV. EXPERIMENTS

Evaluation of the proposed teleoperation system is carried out by: 1) training and testing the SVDD colour classifier; 2) demonstrating some simple teleoperation tasks.

A. *Training and Testing SVDD Classifier*

Since the components of YCbCr can vary greatly due to different illumination. Therefore, images of the 4-ball frame under various environment lighting condition were captured to make the distributions of the four colours, namely blue, green, yellow, red, to obey the actual distributions in real environment.

LibSVM[4] is a well-known SVM tool and is used to train the SVDD classifiers. In this experiments, Gaussian kernel is adopted and its kernel width is set at $\sigma = 3.5$, and the penalty factor is set at $C = 0.001$.

Pictures under different environment lighting conditions are used to demonstrate the validity of the four SVDD classifiers, as is shown in Fig. 8. The experiments above show that the four colour balls can be detected under different illumination condition. In addition, since the speedup techniques, including lookup table method and Kalman filter, are adopted, the vision subsystem can process more than 20 frames per second, which achieves real-time requirement.

B. Some Teleoperation Tasks

In this experiment, the operator holds the four-ball frame to control the UR5. When the teleoperation system starts, the poses of the robot and the feature frame are recorded as their initial poses. The offset pose of the frame is the difference between its current pose and the initial pose. When the robot receives the pose offset of the four-ball frame, it evaluates its target pose by adding the offset pose to its initial pose so that the robot motion can duplicate the operator hand motion.

In this experiment, the operator held the feature frame to do some simple actions and the UR5 followed the action of the operator at the same time. It is demonstrated in Fig. 9 that the pose of the effector of UR5 is the same as the pose of the feature frame. When the feature frame moving, the UR5 can also do the same action. By the aid of vision system, the operator can make some corrections when operating UR5 remotely.

Fig. 9. Teleoperation of the UR5

V. Conclusion

In this paper, we have presented a robot teleoperation system that consists of a 3D vision subsystem and a Universal Robot 5 (UR5), which are interconnected though LAN. The vision system evaluates the position and orientation of a four-ball feature frame held by the operator. Support vector domain description (SVDD) is adopted to train four colour classifiers, based on which the balls on the feature frame can be detected. In order to reduce computation, a novel colour table is proposed and applied to colour detection. Kalman filter is also utilized in the vision system to track the balls of the feature frame to reduce the detection regions. Experiments on the SVDD colour classifiers and teleoperation system are then conducted to validate the good performance of proposed teleoperation system.

Acknowledgment

This study is funded by Beijing Municipal Commission of Science and Technology (Z171100000817009)

References

[1] Mayez A Al-Mouhamed, Onur Toker, and Abdulkhalik Al-Harthy. A 3-d vision-based man-machine interface for hand-controlled telerobot. *IEEE Transactions on Industrial Electronics*, 52(1):306–319, 2005.

[2] Eric R Bachmann, Robert B McGhee, Xiaoping Yun, and Michael J Zyda. Inertial and magnetic posture tracking for inserting humans into networked virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 9–16. ACM, 2001.

[3] Gary Bishop and Greg Welch. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[5] Sokho Chang, Jungtae Kim, Insup Kim, Jin Hwan Borm, Chongwon Lee, and Jong Oh Park. Kist teleoperation system for humanoid robot. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 2, pages 1198–1203. IEEE, 1999.

[6] Guanglong Du, Ping Zhang, Jianhua Mai, and Zeling Li. Markerless kinect-based hand tracking for robot teleoperation. *International Journal of Advanced Robotic Systems*, 9(2):36, 2012.

[7] Max Fischer, Patrick van der Smagt, and Gerd Hirzinger. Learning techniques in a dataglove based telemanipulation system for the dlr hand. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1603–1608. IEEE, 1998.

[8] Tatsuya Harada, Tomomasa Sato, and Taketoshi Mori. Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed. In *Human Motion, 2000. Proceedings. Workshop on*, pages 99–106. IEEE, 2000.

[9] Parham M Kebria, Saba Al-Wais, Hamid Abdi, and Saeid Nahavandi. Kinematic and dynamic modelling of ur5 manipulator. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 004229–004234. IEEE, 2016.

[10] Kengo Koara, Atsushi Nishikawa, and Fumio Miyazaki. Contour based hierarchical part decomposition method for human. *Human Friendly Mechatronics*, page 235, 2001.

[11] Futoshi Kobayashi, Keiichi Kitabayashi, Hiroyuki Nakamoto, and Fumio Kojima. Hand/arm robot teleoperation by inertial motion capture. In *Robot, Vision and Signal Processing (RVSP), 2013 Second International Conference on*, pages 234–237. IEEE, 2013.

[12] Jonathan Kofman, Xianghai Wu, Timothy J Luu, and Siddharth Verma. Teleoperation of a robot manipulator using a vision-based human-robot

interface. *IEEE transactions on industrial electronics*, 52(5):1206–1219, 2005.
[13] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference on Computer Vision*, pages 3–19. Springer, 2000.
[14] Sungman Park, Yeongtae Jung, and Joonbum Bae. A tele-operation interface with a motion capture system and a haptic glove. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on*, pages 544–549. IEEE, 2016.
[15] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.