2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Teleoperation of a Virtual iCub Robot under Framework of Parallel System via Hand Gesture Recognition

Chen Li, Hongbin Ma*, Chenguang Yang*, Menyin Fu

***Abstract*—This paper describes our preliminary development of a virtual robot teleoperation platform based on hand gesture recognition using visual information. Hand gestures in images captured by a camera are recognised to control a virtual iCub. We employ two methods to realise the classification: Adaptive Neuro-fuzzy Inference Systems (ANFIS) and Support Vector Machines (SVM). We realise the teleoperation of a virtual robot using iCubSimulator. The technique in the paper will enable us to teleoperate a physical robot in the future work. In addition, a video server is set up to monitor the real robot. By using the parallel system we are able to improve the robot's performance. Based on the techniques presented in this paper, the virtual iCub can perform the specified actions remotely in a natural manner.**

I. Introduction

A. Background

The rapid development of robot technology integrates various fields including computer technology, control theory, mechanism theory, information sensor technology, artificial intelligence, bionics and other disciplines. Following the great success of industrial robots, social robots have now attracted increasing attention in the last decade [1]. Human-robot interaction (HRI) is an advanced technology and plays an increasingly important role in social robot application. Hand gesture recognition provides us an innovative, natural and user friendly way which is familiar to our human beings. With the latest advances in the fields of computer vision, image processing and pattern recognition, vision-based hand gesture classification in real time becomes more and more feasible for human-computer interaction. Therefore robots can perform tasks based on gesture information in accordance with operator's intension[2].

It is believed that future robots are going to be more interactive and to play more significant roles in our society. The teleoperation technology with the interaction between human and robot has been widely investigated [3]. Teleoperation robot system through the Internet enables us to control a remote robot conveniently, but is subject to some possible disadvantages, such as large signal transmission delay or serious data loss due to the limitations of bandwidth and network transmission protocol [4]. All of these may seriously affect the working performance and may reduce the stability.

In this paper, we employ the framework of a parallel system to improve the performance of the teleoperation system. "Parallel system" refers to the common system composed of a certain real natural system and a virtual artificial system. It is noted that parallel system plays an important role in control and computer simulation systems [5]. With the development of information technology and the increasing availability of networks, many problems involved in the management and control of complex systems could be solved by further investigation with the potential of artificial systems in parallel systems. This helps artificial systems play a better role in the management and control of actual system. In our telerobot control design, the framework of parallel systems could help us to compensate for the effects caused by the uncertainty of the complex system model.

B. Preliminary

We carry out our work based on the virtual iCub Robot. The physical iCub is a one meter tall humanoid robot, which is mainly used to study human's perception and cognition ability. The virtual iCub robot is simulated in the opensource simulator, iCubSimulator, which also provides built-in virtual environment for the robot to interact with [6]. Moreover, its design exactly follows the physical robot, such that results obtained from the simulator can be directly tested on the real robot. In this paper, we use iCubSimulator to achieve human-robot interaction and teleoperation.

Yet Another Robot Platform (YARP) [7] is the communication tools for the robot system which is composed of libraries, protocols and tools. The modules and the devices are separate without interfering each other. Communication with the robot is implemented via YARP.

A dual arm iCub manipulator model built by RVC toobox [8] in MATLAB has been employed. The RVC toolbox provides many functions which are useful for the study and simulation of classical arm-type robotics, e.g. such as kinematics, dynamics, and trajectory generation.

C. Main work

An overview of the teleoperation platform developed in this work is illustrated in Fig.2. We develop the hand gesture recognition system described in Section II. The gesture recognition system based on vision provides us with a convenient and efficient means to control the virtual robot. Parallel systems

This work was supported in part by EU Grant PIIFR-GA-2010-910078, the Royal Society research grant RG130244, the National Nature Science Foundation in China (NSFC) under Grants 61004059 and 61203074, NSFC-RS Joint Project grant (61211130359 and IE111175), and Beijing Outstanding Talents Programme (2012D009011000003).

* To whom all correspondences should be addressed. Tel: (+86)-10-68913985, E-mail: mathmhb@bit.edu.cn;cyang@ieee.org.

The authors are with School of Automation as well as State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing, China, 100081; Chenguang Yang is also with School of Computing and Mathematics, Plymouth University , UK, PL4 8AA.

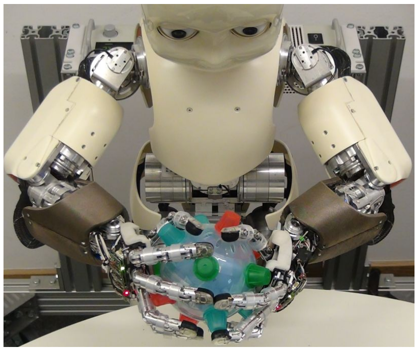


Fig. 1. The iCub Robot (Photo taken in Plymouth University)

theory, described in Section III, is applied into robot teleoperation platform to achieve the interaction between the remote and the local one, as summarized in Section IV. In addition, in order to monitor the performance of robot in real time, a video server system has been set up.

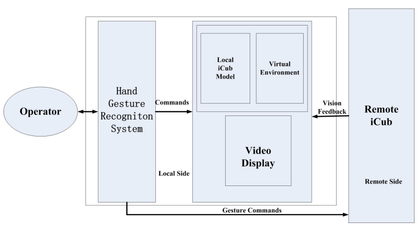


Fig. 2. The overall framework of the system

II. HRI via Hand Gestures

For convenient and natural communications between human and robot, a lot of devices that can detect body motion, hand gestures, sound and speech, facial expression and other aspects of human behaviors can be used. In order to interact with computer naturally, we adopt a hand gesture recognition system based on vision. Gestures are able to convey meaningful information efficiently.

Hand gestures can be captured through a variety of sensors, e.g. “Data Gloves” that precisely record every digit’s flex and abduction angles, and electromagnetic or optical position and orientation sensors for the wrist. In fact, wearing gloves or trackers may be not comfortable and usually increases the “time-to-interface” or setup time. In comparison, computer-vision-based interfaces offer unencumbered interaction, providing a number of notable advantages, e.g. it is non-intrusive, passive and quiet. Installed camera systems can perform other tasks in addition to hand-gesture interfaces; sensing and processing hardware is commercially available at low cost [9].

Vision-based control has drawn considerable attention from researchers in recent years [10]. Gestures are the schematic movement of the hands which can express thoughts or convey orders. Based on computer vision, hand gesture has broad range of applications[11] as a new input modality. An automatic hand gesture recognition system will find many applications in area of human-computer interaction or human-robot interaction. Here we design our control scheme of the virtual iCub robot using iCubSimulator through hand gesture recognition.

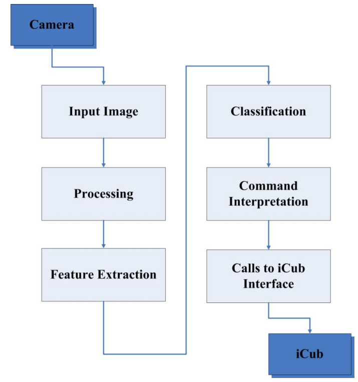


Fig. 3. The hand gesture recognition process

A. Image Processing

Image processing is necessary for feature extraction and classification. It consists of the following steps:

a) Filtering: This step is employed to reduce the noise in the acquisition process and to enhance the images quality. An eight neighborhood filtering is applied to the image to reduce the noise.

b) Segmentation: The image is segmented into two regions: (i) hand gesture region, and (ii) background region. In RGB colour space images are more sensitive to different light conditions so we convert RGB images to YCbCr images, which is a commonly used type of colour space. Y is the brightness component, while Cb and Cr are blue and red concentration offset components. Skin colour will not change with position, size or the direction of hands. So skin colour detection has great versatility in the gesture segmentation.

It is noted that colour distribution is in concordance with the elliptical distribution on Cb’Cr’ plane, while Cb’Cr’ is obtained by nonlinear piecewise colour transformation of CbCr considering the Y component [12]. The Y component will affect the shape of skin colour region. The proposed elliptical clustering model through trial can be mathematically described by

$$\frac{(x-ecx)^2}{a^2}+\frac{(y-ecy)^2}{b^2}=1 \tag{1}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} Cb'-cb_0 \\ Cr'-cr_0 \end{bmatrix} \tag{2}$$

where $cb_0 = 109.38$, $cr_0 = 152.02$, $\theta = 2.53$, $ecx = 1.60$, $ecy = 2.41$, $a = 25.39$, and $b = 14.03$ are computed from the skin cluster in the Cb’Cr’ space.

The pixel belongs to the skin of the hand if $D(Cb, Cr)$ is 1.

$$D(Cb, Cr) = \begin{cases} 1 & \frac{(x-ecx)^2}{a^2}+\frac{(y-ecy)^2}{b^2} \le 1 \\ 0 & \text{others} \end{cases} \tag{3}$$

c) Boundary extraction: Boundary information is very important for feature extraction. Determining the border of the gesture region is the last step in the image processing phase. We use MATLAB command “bwboundaries” to extract contour of the quasi-binary image. Then, we get binary images and extract the hand gesture contour.

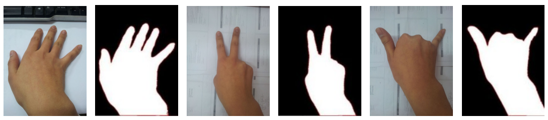


Fig. 4. Three gestures and their binary images

B. Features Extraction

In order to improve the robustness of recognition, we use more than one feature to describe the gestures.

There are mainly three groups of features to be detailed in the following: Hu moments, Fourier descriptors, gesture regional features.

d) Hu moments: Hu introduced seven nonlinear functions defined on regular moments [13]. These moments are used to reflect the distribution of random variables in statistics. Extending to the mechanics, they can be used as a quality distribution characterization. For the characteristics invariant to translation, scale, and rotation, Hu invariant moments are employed for the representation of hand gesture. The low-order moments mainly describe overall features of the image, such as area, spindle and direction angle; high-order moments mainly describe details of the image, such as twist angle. We use the first five Hu moments, M_1,M_2,M_3,M_4, and M_5.

e) Fourier descriptors: Fourier descriptors[14] are the Fourier transform coefficients of the object boundary curve. We adopted Fourier descriptors to represent the boundary of the extracted binary hand as the set of complex numbers, b_k = x_k+ jy_k; where $\{x_k, y_k\}$ are the coordinates of pixels on the boundary. This is resampled to a fixed length sequence, $\{f_k, k = 0, 1, \cdots\}$, for use with the discrete Fourier transform (DFT). Fourier descriptors will be influenced by scale, direction and starting point of the shape. So, it is necessary to make normalisation for Fourier Descriptors. We use FFT to achieve fast Fourier transform of boundary points in MATLAB. Only the second to eleventh normalized coefficients are used.

f) Gesture regional features: Regional characteristics generally include geometric features (such as area, center and breakpoint), shape features, brightness characteristics (such as average brightness) and various texture statistical features. This paper uses several commonly-used geometry and shape characteristics. We calculate the ratio of actual area of hand to convex hull and the ratio of perimeter squared to area. Convex hull is a convex polygon formed by connecting the outermost points on plane which include all points. Boundary points and the convex figure of gesture “five” are drawn in the axis as shown in Fig.5.

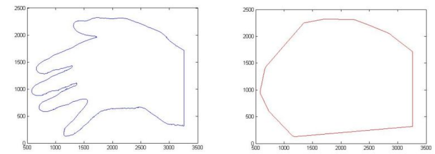


Fig. 5. (a)The contour of gesture five; (b)Its convex hull

C. Gesture Recognition

In order to realize the classification and recognition,we adopt two means: ANFIS and SVM.

g) ANFIS: The adopted Adaptive Neuro-Fuzzy Inference System (ANFIS) architecture is of the type that is functionally equivalent to the first order Sugeno-type fuzzy inference system [15]. Fuzzy inference does not have its own learning mechanism, consequently, it is limited in practical use. Meanwhile, ANNs cannot express hazily. In fact, it is similar to a black box lacking transparency, therefore, it may not reflect the human brain’s functions, such as abstract reasoning. Fuzzy inference based on ANN could combine fuzzy inference and artificial neural network, and preserve the advantages of the two and overcomes the individual shortcomings. Adaptive neural fuzzy inference system provides a learning method to extract information from dataset for the process of fuzzy modelling. Learning in this method is very similar to using a neural network. It can effectively calculate the best parameters of membership functions to make sure that the design of the Sugeno fuzzy inference system can simulate the desired or actual input-output relationship. We use Fuzzy Logic Toolbox in MATLAB to build the adaptive fuzzy neural network. We create an initial fuzzy inference system and combine the hand gesture feature vectors with gesture labels according to a certain format to be the training data. We also train the fuzzy inference system in ANFIS method to adjust the membership function parameters. The FIS model will approximate given training data continuously.

h) SVM: Support Vector Machine is a popular machine learning method for classification, regression, and other learning tasks. LIBSVM is an integrated software for support vector classification [16]. It is developed by Lin Chih-Jen who is an associate professor of Taiwan University. It supports multi-class classification. Here we use the LIBSVM toolbox for gesture classification.

D. Commands to iCub

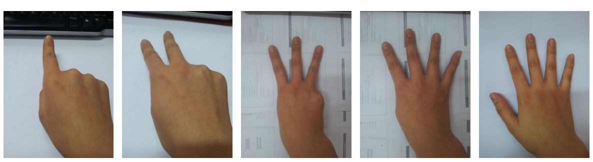
The virtual iCub is controlled with the identified gestures with specific meaning. We move virtual iCub by changing the 16 joints’ angles of the arms and the hands. For example, when we extend our index finger, which is recognized in the picture captured by the camera, the virtual iCub robot will raise his right arm up. We use YARP to set communication channel between MATLAB and iCubSimulator. Similarly to [17], we call YARP by running “LoadYarp” command in MATLAB. Through the YARP interface, we are able to initialize iCubSimulator and set postures of its arms based on the gestures recognized. Then iCubSimulator wil then perform the appropriate action.

E. Experiment

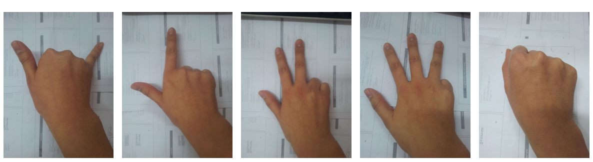
A USB external camera is used to capture photos of hands. 100 samples for each gesture were taken. For each gesture, 70 out of the 100 samples were used for training purpose, while the remaining 30 samples were used for testing. The samples were taken from different distances and orientations. In this way, we are able to obtain a data set with cases that have different sizes and orientations, so that we can examine the efficiency of our feature extraction scheme. Some of the samples taken for the hand gesture of "five" are shown in Fig.5. When the program is running, we shoot a picture every few seconds for testing. The dimension of boundary feature vectors for each image is seventeen including five Hu moments, ten Fourier coefficients and two ratios about gesture regional features.

Fig. 6. Some of the samples taken for the gesture "five"

We use ten gestures representing ten movements for iCub to do.



(a) 1 (b) 2 (c) 3 (d) 4 (e) 5



(f) 6 (g) 7 (h) 8 (i) 9 (j) 10

Fig. 7. The ten basic hand poses making up the system gestures: (a)raise right arm upward; (b)raise left arm upward; (c)raise both arms upward; (d)raise right arm horizontally forward; (e)raise left arm horizontally forward; (f)raise both arms 45 degrees upward; (g)arms swing up and down; (h)bend left arm ; (i)bend right arm; and (j)clenched fist

We list three examples of the obtained feature vectors in Table 1.

Hand gesture recognition results are listed in Table 2 and Table 3.

Commands of "one","three","ten" correlate to the iCub raise its right arm, to raise two arms and to make a fist, as is shown in Fig.8.

TABLE I
FEATURE VECTORS OF THREE EXAMPLES

feature vectors of picture one	feature vectors of picture three	feature vectors of picture ten
1.27344128668080	0.768013917897862	1.15287474965257
0.341034945794739	0.205285988633656	0.231987711436551
0.0345098405446921	0.0551058692738506	0.0200987494896591
0.0885600310397111	0.100618520664892	0.0419426570679495
0.000681031472695372	0.00479048615623522	-3.63390033686653e-05
1013.78223135593	818.120925380823	1051.57474197919
103.045643221326	295.425599120192	45.4506742294877
49.7487851990050	78.2868290117613	74.7276500637026
33.7888356801995	40.6368968772091	41.7750625708967
46.5428205412630	119.211566543840	24.8569400929022
17.7746780455021	67.2151017492618	5.39515370707666
16.7520226903149	23.6122301399800	11.8188665303865
8.91967851920738	24.2924115274574	8.45940554065641
7.67213502927261	10.1949456741707	7.77960672239471
2.65334909384366	15.2443595339593	5.54551570458614
0.896527786365779	0.729194795728057	0.924981631713313
25.1125100024321	61.4839999790745	23.0492411591836

TABLE II
RECOGNITION RESULT WITH SVM

gesture	train number	test number	successful cases	recognition rate (%)
1	70	30	30	100
2	70	30	29	96.67
3	70	30	30	100
4	70	30	28	93.33
5	70	30	30	100
6	70	30	30	100
7	70	30	30	100
8	70	30	29	96.67
9	70	30	30	100
10	70	30	30	100

Fig. 8. (a)Raise right arm;(b)Raise two arms;(c)Make a fist [snapshot]

III. TELEOPERATION PLATFORM AND PARALLEL SYSTEM

The teleoperation system consists of two parts: remote iCub robot and local iCub manipulator model built in MATLAB referred to in Fig.10. We employ a virtual iCub. It is designed the same as the real one. Based on human-computer interaction, we accomplish controlling virtual iCub robot conveniently and interacting with the immersion environment naturally. To facilitate the future research on network time delay problem and for better application, we employ the concept of parallel system. We use a iCub manipulator model built in MATLAB with RVC toolbox locally and install iCubSimulator in the remote side while implementing the local control of iCub manipulator model and the remote control of virtual iCub using iCubSimulator. In future work, the technology developed in this paper will be used. As there are some errors between the simulation model and real iCub movement, such as the virtual robot has touched objects but the real has not, and the reals work environment is invisible, it is not reliable to completely rely on the prediction of simulation that we must know the remote real situation. Therefore, we need to obtain remote video information and accomplish the integration of the models and video.

A. Parallel System

In order to control the robot better, to make the robot evolve better and to realise the remote operation, we employ

TABLE III
RECOGNITION RESULT WITH ANFIS

gesture	train number	test number	successful cases	recognition rate(%)
1	70	30	29	96.67
2	70	30	29	96.67
3	70	30	28	93.33
4	70	30	27	90
5	70	30	29	96.67
6	70	30	30	100
7	70	30	30	100
8	70	30	29	96.67
9	70	30	28	93.33
10	70	30	30	100

the framework of parallel system mentioned above. We adopt the parallel system framework based on virtual reality to optimise the performance of the robot. In virtual simulation environment, the operator can get feedback of the real-time operating results, and the remote robot repeat the simulation results after a certain delay, so the operator can effectively operate continuously facing the operating virtual simulation model with the influence of time delay. The iCubSimulator and the manipulator model form a collaborative intelligent simulation platform. On one hand, iCub manipulator model acts as the virtual model for the real robot while complete the parallel, on-line, real time simulation. On the other hand, iCub manipulator model collaborates with the virtual iCub robot, develop high-level decision-making ability, extend and expand iCub robot's functions.

The pattern that the physical robot and the model work together intelligently to complete real-time simulation is different from the previous traditional one. Traditional simulation is offline, serial, open loop, but the coordinated intelligent simulation is dynamic, closed-loop, real-time, parallel and online (refer to Fig.9).

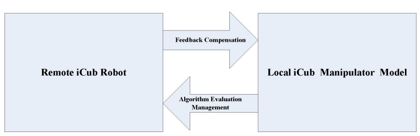


Fig. 9. The parallel system

B. iCub Manipulator Model in MATLAB

Based on the RVC toolbox [8], an iCub manipulator model is set up in MATLAB as shown in Fig.10. RVC toolbox provides powerful support for a lot of classical mechanical arm model on dynamics, kinematics and trajectory planning. The virtual iCub robot model built in MATLAB is simulated in computer. Then we can use hand gesture recognition system to control it.

C. iCubSimulator

We communicate with the virtual iCub robot via YARP. The created ports transfer information through a server called "yarpserver". Then we can realise the remote control of virtual iCub.

First, we must configure both A and B with the command-line command "yarp conf $< ip >$", where $< ip >$ denotes the true IP address of the computers. We should connect the local

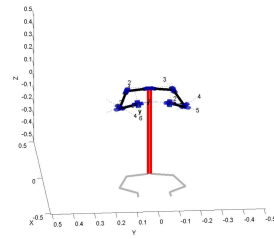


Fig. 10. iCub in MATLAB [snapshot]

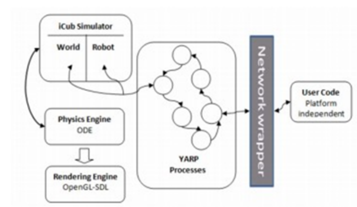


Fig. 11. YARP [17]

YARP output port a and the remote YARP port b. Then we can control the remote robot through sending commands from port a to port b.

For simplicity, we install a virtual machine on local machine. Connect the virtual and the local one. Then we can control iCub on local computer through sending commands from virtual machine as shown in Fig.12.

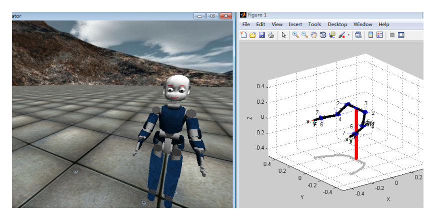


Fig. 12. Two iCubSimulator working together[snapshot]

IV. FURTHER WORK

In order to enable the operator to monitor the movement of the remote robot in real time, we need to feed back information from the remote side. Through the network, live video image information and the robot motion state information may be fed back to the operator side. The video monitor should not only provide information to the operator, but also be fused with the virtual model for later correction of the simulation model, and modify the virtual simulation environment. Then, we can also make the video of local hand gesture transported to remote side which will help to improve security.

There are mainly three steps: camera calibration, acquisition and transmission of video information, overlay of video images and simulation graphics.

A. Camera Calibration

Camera calibration involves several coordinate transformations.

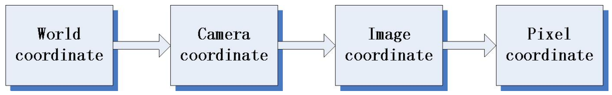


Fig. 13. The coordinate transformation

B. Video Capture and Transmission

On video transmission parts, we choose streaming media technology. VLC is a free, cross-platform, and open-source multimedia player and framework that can play most multimedia files and support a variety of streaming media protocols. It is simple and easy using VLC [18] to build a video server. This allows many users online to see the motion of the real iCub robot.

We put up a streaming video server with VLC, transmiting real-time video captured by the camera. When using VLC to build streaming media server, we can choose the port, encoding mode and the transmission protocols.

We could put the VLC as video server in remote controller and transmit the real robot motion captured as real-time video streaming. We put the VLC as client in local place to play the real-time video streaming.

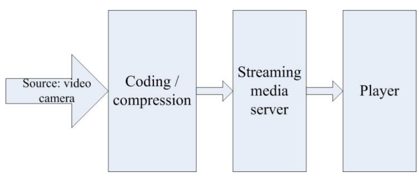


Fig. 14. Streaming media network diagram

On the local side, MATLAB can call VLC ActiveX control, so the real-time video can display in MATLAB window. VLC can also preserve the received real-time video streaming for facilitating the real-time processing.

In future work, we can fuse simulation graphics and video images.

V. Future Work

In future work, we are going to implement augmented reality using video fusion. In the simulated environment, we calculate the calibration matrix of camera according to the camera calibration algorithm and then multiply the matrix with the arm points of iCub model in MATLAB [19]. Two-dimensional pattern can be obtained by perspective projection with the orthogonal projection onto the two-dimensional graphic image plane. Then iCub manipulator model can be superimposed on the image with graphics and images overlap in the same window, which benefits easy comparison of ... For ease of comparison, virtual simulation model can be simply chosen as a wireframe model.

VI. Conclusion

In this work, we have established a basic robot teleoperation platform with a hand gesture recognition system based on vision information. The features selected are particularly useful for real-time control and able to enhance robustness. We make it easy to control a virtual iCub robot with gestures. The vision-based hand gesture recognition system has met the performance requirements including real-time performance, accuracy, and robustness. Based on the hand gesture recognition, the teleoperation of virtual iCub has been realized. It may help to manipulate a robot remotely. A video server has been set up as further work, and in this way we can monitor the remote scene in real time, which may help to control the remote robot by adaptive prediction and compensation using the parallel system. In future work, we will investigate more methods of adaptation and learning under a realistic scenario of teleoperation robot system.

Acknowledgement

The authors thank Alex Smith for his help to build the iCub robot model in MATLAB with RVC toolbox and to improve the English of the paper.

References

[1] B. R. Duffy. Anthropomorphism and the social robot. *Robotics and autonomous systems*, 42(3):177–190, 2003.
[2] C. Qing, N. D. Georganas, and E. M. Petriu. Real-time vision-based hand gesture recognition using haar-like features. *Instrumentation and Measurement Technology Conference*, 2007.
[3] N. Hogan. Controlling impedance at the man/machine interface. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 1626–1631. IEEE, 1989.
[4] K. Brady and T. J. Tarn. Internet-based remote teleoperation. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 65–70. IEEE, 1998.
[5] F. Y. Wang. Parallel system methods for management and control of complex systems. *Control and decision*, 19:485–489, 2004.
[6] Wiki for the iCub simulator specifications from its installation to its use. http://www.eris.liralab.it/wiki/.
[7] Yet another robot platform. http://yarp0.sourceforge.net/.
[8] P. I. Corke. Robotics toolbox. 2008.
[9] J. P. Wachs, K. Mathias, S. Helman, and E. Yael. Vision-based hand-gesture applications. *Communications of the ACM*, 54(2):60–71, 2011.
[10] K. Hashimoto. A review on vision-based control of robot manipulators. *Advanced Robotics*, 17(10):969–991, 2003.
[11] P. Meenakshi. Hand gesture recognition based on shape parameters. *International Conference on Computing, Communication and Applications*, pages 1–6, 2012.
[12] Re. L. Hsu, A. M. Mohamed, and A. K. Jain. Face detection in color images. *Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.
[13] M. K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.
[14] G. H. Granlund. Fourier preprocessing for hand print character recognition. *Computers, IEEE Transactions on*, 100(2):195–201, 1972.
[15] N. Nadia (ed.). Adaptation of fuzzy inference system using neural learning, fuzzy system engineering: Theory and practice. *Studies in Fuzziness and Soft Computing*, pages 53–83, 2001.
[16] C. C. Chang and C. J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
[17] P. Renon, C. G. Yang, H. B. Ma, and R. X. Cui. Haptic interaction between human and virtual iCub robot using novint falcon with chai3d and matlab. *Chinese Control Conference*, 2013.
[18] VLC. http://www.videolan.org/.
[19] G. C. Zhu, T. M. Wang, W. S.Chou, and M. Cai. Research on augmented reality based teleoperation system. *Acta Simulata Systematica Sinica*, 5:021, 2004.