Interactive Teleoperation Interface for Semi-Autonomous Control of Robot Arms

Camilo Perez Quintero, Romeo Tatsambon Fomena, Azad Shademan, Oscar Ramirez and Martin Jagersand*

***Abstract*—We propose and develop an interactive semi-autonomous control of robot arms. Our system controls two interactions: (1) A user can naturally control a robot arm by a direct linkage to the arm motion from the tracked human skeleton. (2) An autonomous image-based visual servoing routine can be triggered for precise positioning. Coarse motions are executed by human teleoperation and fine motions by image-based visual servoing. A successful application of our proposed interaction is presented for a WAM arm equipped with an eye-in-hand camera.**

I. Introduction

Structured environments and repetitive tasks made robots succeed in industry. Unfortunately, human environments are unstructured, dynamic and normally require human interaction. Robotics researchers and robot companies have been struggling for more than four decades to bring robots closer to humans with only few finely tailored-task successes. In Japan's Fukushima Daiichi power plant disaster, a highly unstructured and unpredictable environment, weeks passed before power plant personnel completed training to operate the few available rescue robots [1]. Typically, robots are instructed either by text-based programming or direct control of motions. Learning to teleoperate a multiple-DOF robot is cumbersome and time consuming. Human environments require more natural communication mechanisms that allow humans to effortlessly interact with a robot.

Several approaches have been explored for unstructured environments: For instance, a forcefeedback device is used to teleoperate a humanoid robot [3]. This incurs an increased cost as highly specialized hardware is required. Grasping through visual interfaces is another approach [4], but objects outside of the field of view are not easily accessible. Kofmal *et al.* [2] present a vision-based method of robot teleoperation that allows a human to communicate simultaneous DOF motion to a robot arm by having the operator perform 3D human hand-arm motions. However, markers and special backgrounds are required which would not be available in a real environment.

We propose and explore a hybrid system which allows users to execute coarse motion teleoperation and, if precise motions are required, a visual servoing routine is launched with a gesture interface. Our system does not need special background or markers in the environment. Furthermore, our system relieves users from precise motions by introducing an autonomous routine (visual servoing). Fig. 1 shows our system. By using an intuitive gesture interface, the user in Location 1 tracked by the Kinect is able to teleoperate the robot arm in location 2. Our proposed HRI combines the strengths of both teleoperation and visual servoing. For large motions tele-operation is quicker than visual servoing, since the eye-in-hand camera has a limited field of view and it would be tedious and unintuitive for the user to define several segments of visual servoing. For precise manipulation on the other hand the direct mapping of tracked human arm motions to robot motions suffers from noise in the tracking and it is difficult for the human to deal with the dynamics of the robot. The end result is that teleoperated motions, while fast, are jittery and not very precise. Here visual servoing helps the human by relieving him from dealing with the dynamics of the robot, and allowing very precise motions.

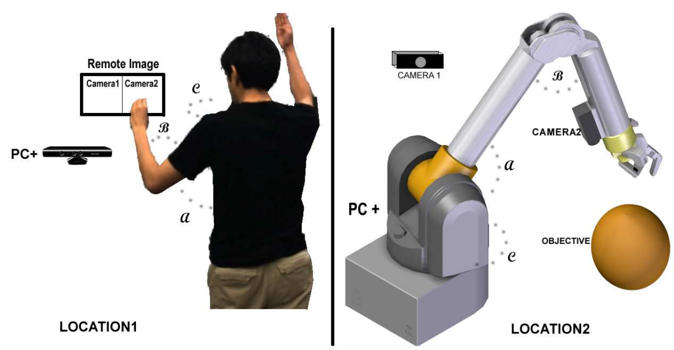

Figure 1. Left: A left-handed user gestures and arm joints are tracked with a Kinect. Right: Robot equipped with eye-in-hand and eye-to-hand camera. The system can be controlled either coarsely and quickly through teleoperation by replicating the users arm movements, or with high precision but slower though visual servoing towards visual goals defined by the operator.

In the next section we present the fundamentals of teleoperation and visual servoing, which are the basis of our system. Then, in Section III we describe our system from a hardware, software and user interface points of view. Finally, in Section IV we present experimental results that validate our chosen design.

*This work is supported by NSERC and the Canadian Space Agency (CSA).

Authors are with the Department of Computing Science, University of Alberta, Edmonton, AB, T6G2E8, Canada. caperez@cs.ualberta.ca

II. Background

In this section we review the fundamentals of the two control modes used as the basis of our system. The teleoperation mode which allows for coarse robot arm motions and the automonous mode which allows for precise robot arm motions.

A. Teleoperation

The use of tele-manipulation precedes the currently common use of robotics in automated manufacturing. The first tele-manipulators were pure mechanical linkages designed to distance the human operator from hazardous objects. Examples include the manipulation of radioactive materials from behind a radiation-proof window [12].

Electric-drive tele-manipulation came later and such systems commonly use a conventional robot arm as a slave device, while the human master motion interface can take a variety of forms. On the positive side, these systems allow much larger separation between the human operator and slave robot, enabling applications from e.g. police bomb defusing to space tele-robotics [12]. On the negative side, complex systems are often difficult to operate.

On the surface, it may seem that tele-manipulation should be easy. Motions sensed by the master device need simply be replicated by the slave robot. However, a higher mass of the robot arm and hand, combined with limitations in control, contributes to a dynamic response of the robot that is different from regular human manipulation. Small delays in transmission and slight imperfections in the system (e.g. a small dead zone) further add to the difficulty.

Despite these challenges tele-manipulation or combinations of tele-manipulation and autonomous control (e.g. supervisory control [14]) have promise in many unstructured robotics tasks where full autonomy is not possible or not desirable. Recently research in tele-manipulation has addressed technical issues such as the stability of a master-slave system under delays [16], issues with visual and haptic rendering of feedback to the operator including registration of models with video, communication delays [15] and better operator interfaces and geometric mappings [13].

Searching for a more intuitive approach our teleoperation consists in a direct linkage between the human arm and the robot arm as shown in Fig. 1. Although at first having elbow up and elbow down configuration, for the robot and the human respectively, seems not so intuitive, we have found that the user centers more his attention on the end effector position than on the rest of the arm. In particular we chose this configuration to avoid hitting obstacles in the robot workspace (e.g. table). The direct linkage allows the user to teleoperate the robot when coarse motions are needed, and for precise motions we have implemented and autonomous control routine based on uncalibrated visual servoing, which is now a well-established robots control technique.

B. Uncalibrated Visual Servoing

Visual servoing consists of using feedback provided by one or several vision sensors to control the motion of a dynamic system [17]. In the case where the model of the system to control is available, calibrated visual servoing approaches can be used. On the other hand uncalibrated visual servoing (UVS) studies vision-based motion control of robots without using the camera intrinsic parameters, the calibration of the robot-to-camera transformation, or the geometric object/scene models [18], [23]. This is a demanding problem with increasing applications in unstructured environments, where no prior information is provided [19], [23], [20], [21].

The control law in the UVS should be defined without the need to reconstruct the depth or other 3D parameters. One way to define the uncalibrated control law is an approach similar to the image-based visual servoing (IBVS). Let $F : \mathbb{R}^N \to \mathbb{R}^M$ be the mapping from the configuration $\mathbf{q} \in \mathbb{R}^N$ of a robot with N joints, to the visual feature vector $\mathbf{s} \in \mathbb{R}^M$ with M visual features. For example, for a 6 degrees of freedom (DOFs) robot with 4 point features (8 coordinates in total), $N = 6$ and $M = 8$. The visual-motor function of such vision-based robotic system can be written as

$$\mathbf{s} = \mathbf{F}(\mathbf{q}). \tag{1}$$

This formulation is general and covers both the eye-in-hand and eye-to-hand systems.

The time derivative of the visual-motor function in (1) leads to

$$\dot{\mathbf{s}} = \mathbf{J_u}(\mathbf{q})\dot{\mathbf{q}}, \tag{2}$$

where $\dot{\mathbf{q}}$ is the control input and $\mathbf{J_u} = \frac{\partial \mathbf{F}(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{M \times N}$ is called the *visual-motor Jacobian*. The discrete-time approximation of (2), when $\mathbf{J_u}(\mathbf{q})$ is replaced by $\widehat{\mathbf{J}}_\mathbf{u}(\mathbf{q})$ is

$$\Delta\mathbf{s} \simeq \widehat{\mathbf{J}}_\mathbf{u}(\mathbf{q})\dot{\mathbf{q}}. \tag{3}$$

Similar to the IBVS control law, the estimated visual-motor Jacobian, $\widehat{\mathbf{J}}_\mathbf{u}$, appears in the uncalibrated control law:

$$\dot{q} = -\lambda \widehat{\mathbf{J}}_\mathbf{u}^{+}(\mathbf{s} - \mathbf{s}^*), \tag{4}$$

where $\widehat{\mathbf{J}}_\mathbf{u}^{+}$ is the Moore-Penrose pseudoinverse of $\widehat{\mathbf{J}}_\mathbf{u}$ and s^* is the vector containing the desired values of the features.

In the control law (4), the visual-motor Jacobian $\widehat{\mathbf{J}}_\mathbf{u}$ is estimated from data. Different methods of estimation exist, for example the orthogonal exploratory motions method [22], the Broyden method [23], [24], the least-squares based method [19]. In this paper we have chosen the Broyden method for its simplicity. This method can be summarized as follows

$$\widehat{\mathbf{J}}_\mathbf{u}^{(k+1)} = \widehat{\mathbf{J}}_\mathbf{u}^{(k)} + \alpha \frac{\left(\Delta\mathbf{s} - \widehat{\mathbf{J}}_\mathbf{u}^{(k)}\Delta q\right)}{\Delta q^\top \Delta q}, \tag{5}$$

where α is the forgetting factor which is used to lessen the weight of old data during the estimation process. The initial guess $\widehat{\mathbf{J}}_{\mathbf{u}}^{(0)}$ of the visual-motor Jacobian can been estimated using orthogonal exploratory motions method.

In our system we implemented a simple uncalibrated visual servoing scheme where s is the center location of our camera (green and yellow ring in Fig. 2) and s^* is the tracking location of the objective (red and blue ring in Fig. 2).

Figure 2. User views. Left: Eye-to-hand camera. Right: Eye-in-hand camera.

III. System Description

Here we first present the user interface and the system state machine. Then we present the system hardware and software. A demonstration of our system can be seen in our website:
http://webdocs.cs.ualberta.ca/%7evis/HRI/teleopVS.wmv

A. Operator Interface and System State Machine

In the tele-operation mode, the user's dominant hand pose is tracked and sent as a position command to the robot. We have put some efforts in designing a suitable geometric mapping between the human hand position and the robot joint angles (Fig. 1), and reasonable control gains for the human to feel that the robot response is natural. Despite this there is an inevitable effect of tracking jitter and robot dynamics which makes precise tele-manipulation difficult. In the visual servoing mode (autonomous control mode) the operator left arm is used for deictic pointing commands (Fig. 3). The human points to a visual target, and visual servoing is initiated to this goal.

Fig. 4 shows our system's state machine design. It consists of a starting state and four states or modes of operation. The human operator gestures with one arm to switch between states/operation modes. The other arm is used to provide spatial information. The system permits to arbitrarily set-up right or left handed mode. Here the system is described for a left handed, the reverse would be used for a right handed.

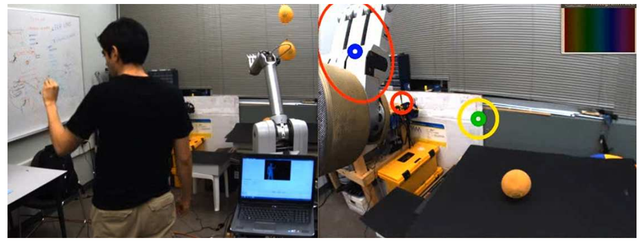

Figure 3. User's left hand, controls the red ring cursor which allows the user to select an object in the image space. Blue ring and red ellipse indicate the currently tracked region. Yellow ring and green ring shows the center of the camera.

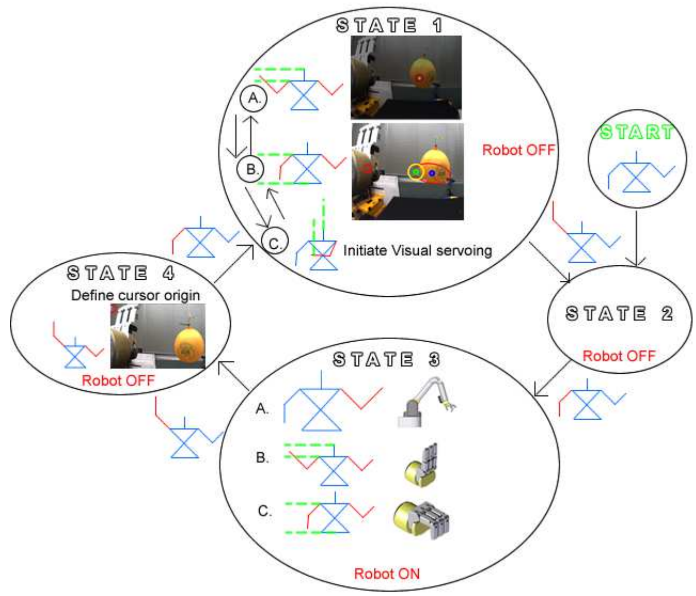

Figure 4. State Diagram. The user shift through states by raising his non dominant hand. Teleoperation is inside *State 3* and the autonomous visual servoing is inside *State 1*. Notice that the user in the diagram is facing out the paper (left hand user).

The system is initialized in the *Start State* (top right Fig. 4). The user arms are pointing down, and the Kinect initializes the skeleton calibration of the user, while the robot is off. Then the user raises his right hand and the system shifts to *State 2* (notice that the user representation in Fig.4 is for a left hand user facing out the paper). The robot is still off, the user locates his left hand in a desire initial position, when the user lowers his right hand the system changes to *State 3*. The robot is on, and in direct linkage to the users left hand. In this state besides the teleoperation of the robot-arm, the user can control the opening and closing of the robot-hand by positioning the user right hand between his shoulder and his head (see *B* and *C* inside *State 3* Fig. 4). For shifting to *State 4* the user raises his right hand over his head. In this state the robot turns off and the user defines a initial mapping of his left hand 3D world position to a cursor situated in the left corner of the eye-in-hand camera

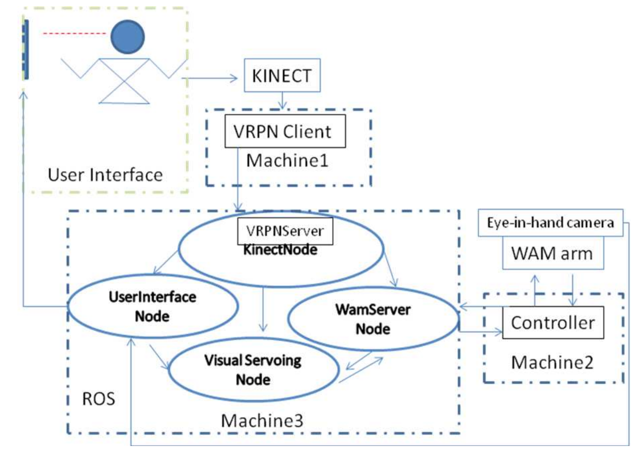

Figure 5. System Architecture. A visual interface is presented to the user to interact with the system. The system receives input signals from the user tracked movements. Machine 1 is located in the users location while, machine 2 and 3 in the remote site with the robot.

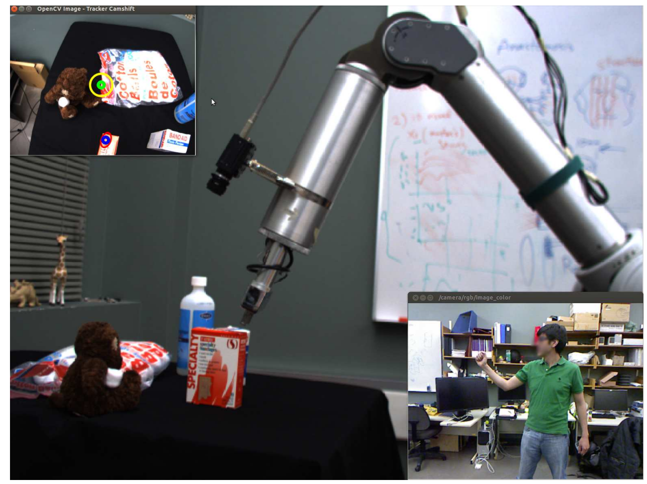

Figure 7. Direct teleoperation is used for getting close to a target box. Then by gesturing the user shifts to autonomous visual servoing mode to center the target in his field of view.

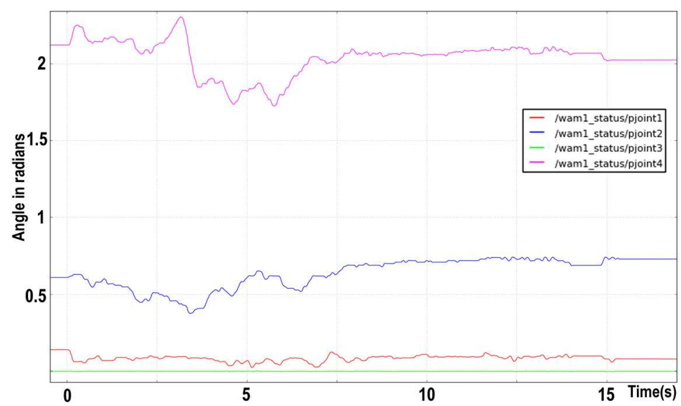

Figure 8. Robot joints behaviour during direct teleoperation for getting close to the target. Joints 1, 2 and 4 correspond respectively to *e, a, B* in location 2 Fig. 1.

(small red circle in Fig. 2). After lowering his right hand the system shifts to *State 1*. The 2D cursor inside the eye-in-hand camera image coordinate space is controlled by a direct mapping to the user's left hand position. *State 1* has an internal state machine composed by *A*, *B*, *C* internal sub-states (see Fig. 4 *State 1*). *A*: The user locates his right hand between his shoulder and top head, the system selects a region of interest based on the 2D cursor location. *B*: The user locates his right hand below his shoulder, a Cam-Shift algorithm [25] runs for tracking the selected object. *C*: A visual servoing routine is activated and the robot moves autonomously to a position where the object of interest is centered in the eye-in-hand camera field-of-view.

B. Hardware and Software Infrastructure

Our system consists of a Kinect sensor and a Windows machine in location 1; a WAM arm, two Linux machines and two cameras in location 2 (see Fig. 1). Fig. 5 shows our system architecture. We have used FASST [5] a middleware implementation that incorporates a VRPN [7] server for streaming the user skeleton joints, read by the Kinect, over a network (see Fig. 5 VRPN client), the motivation for using a windows machine in the user side is mainly for making the system available to a broad range of users.

We have used the open source Robot Operating System ROS [8] to facilitate our system integration. We have designed four ROS components (called nodes): *Kinect node* implements a VRPN client which allows the system to read the skeleton data and map the joints values to the robot joints. *WamServer* node which is in charge of updating the robot position. *User interface* node that reads the user 3D hand position from the Kinect node and converts it into a 2D image position which permits the user to interact in the eye-in-hand 2D image space. The *visual servoing* node is in charge of managing autonomous robot motion routines (see Fig. 5).

IV. Experiments

We performed two sets of experiments. The first set aims at showing a comparison between the system in teleoperation mode and the system switching behaviour between the teleoperation mode and the autonomous mode. The second set illustrates the system noise and delay.

The first experiment consists in locating the arm gripper in a suitable position for grasping a specific object, Fig. 7 shows the task. This was done twice: first the user controls the robot only through teleoperation. Then in the second trial the user teleoperates through coarse motions and autonomous visual servoing takes over control for fine motions. Figure 8 plots the joint values of the robot as it was teleoperated during the first trial.

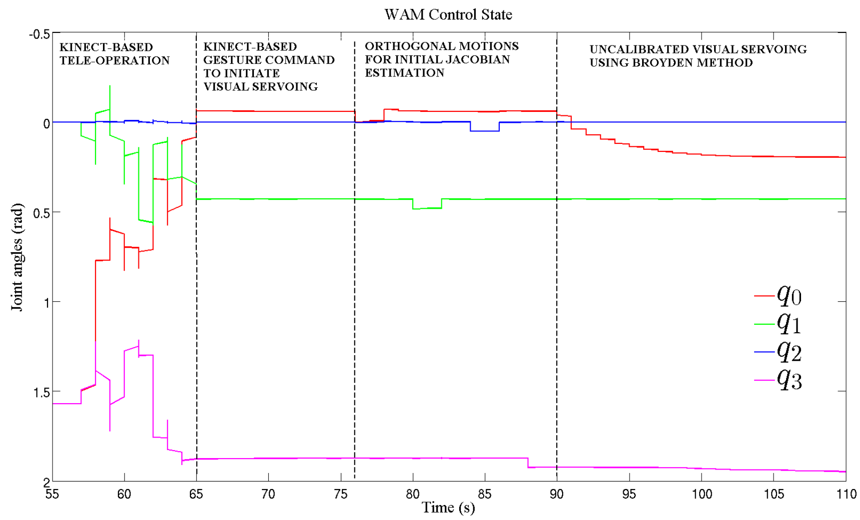

Figure 6. Robot teleoperation and visual servoing routine. Joints q0, q1 and q3 correspond respectively to *e, a, B* in location 2 Fig. 1.

Fig. 6 plots the robot joint values during the second trial. This figure is divided into four periods of time. The *Kinect-based teleoperation* period corresponds to State 3 in Fig. 4. During this period the three joint angles are linked to the human arm. The plot shows a jittering behaviour leading to increased difficulty when attempting precise motions. The *Kinect based gesture command to initiate visual servoing* period shows the stage where the user interacts with the gestures interface, initiating the visual servoing routine. The user selects the desired target and the Cam-Shift algorithm is automatically chosen to track the object of interest.

The *Orthogonal motions for initial Jacobian estimation* period shows the small movements performed by each joint for generating the initial Jacobian. The *Uncalibrated visual servoing using Broyden method* period shows how a numerical method is used to update the Jacobian until the system reduces the error between the tracker location and the center of camera location in the eye-in-hand camera.

In order to show how the depth camera and robot dynamics affect our system, we design three simple qualitative experiments. In all the experiments the user's 3D hand coordinates are obtained by the Kinect and then mapped into a 2D image as the (u, v) cursor. It is shown as a red circle in Fig. 9.

For having a visual feedback of the robot-arm controller and the sensor reading the user hand (y, z) position is converted into robot joint values. At the same time we are reading the actual robot joints and turn them back to y and z coordinates which are then mapped into the (u, v) cursor coordinates, this give us feedback of the system behaviour.

In experiment 1, the user is asked to move a cursor and hold it for 30 seconds in a fix position (see Fig. 9 left upper corner). Although the user hand is static, a noisy response of system is noticeable as shown in Fig. 10.

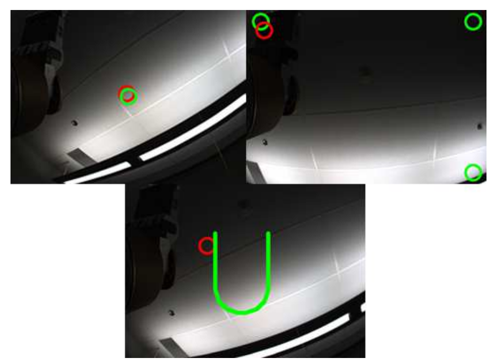

Figure 9. Upper left corner shows reference center point in green and user cursor in red. Upper right corner shows three reference points in green and user cursor in red. Bottom shows the reference curve pattern in green and user cursor in red.

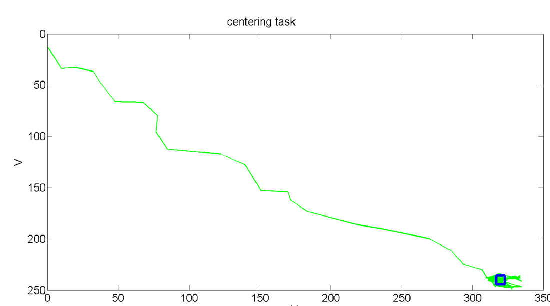

Figure 10. Fixed position experiment drift. The user moves the cursor from top-left corner to bottom-right corner and hold.

In Experiment 2, the user is asked to move cursor in

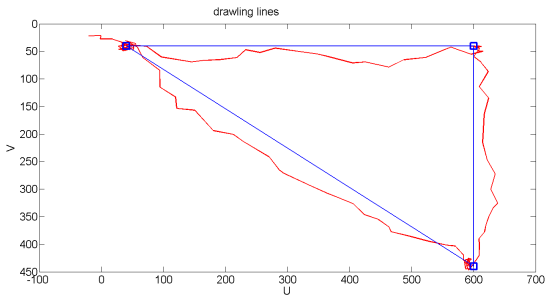

Figure 11. The user moves the cursor in straight line movements, red line shows user performance.

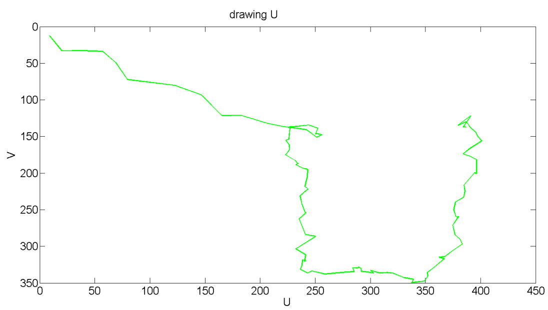

Figure 12. The user moves the cursor in a curved movement.

straight lines from target to target and wait approximately 5 seconds in each one. Fig. 11 shows the user performance during the straight line test evaluation. In experiment 3, the user is asked to follow a curved pattern. Fig. 12 shows the user performance during the curve motion evaluation. The divergence from the edges on the drawing reflect the unprecise motion of the robot. This can be explained by human error along with the noise introduced in the system by the depth sensor and the delay caused by robot dynamics. In average the system presents an error of ± 12 pixels. that means approximately in the users view eye-in-hand camera (see Fig. 2) an error of ± 8cm.

These results validate our system design and confirm the choice made of using teleoperation for coarse motions, and visual servoing for precision motions. In particular for visual servoing, it has been demonstrated that visual control positioning can be more accurate than robot joint control positioning [23].

V. Conclusion

We have proposed a hybrid system which allows the user to control a robot arm with two modes of operation.
We have designed an intuitive gesture interface for switching between the two modes. Mode 1 allows the user to teleoperate a robot arm directly from the user arm movements and with some gestures control commands like hand opening and closing. Mode 2 allows the user to select an object of interest, initiate a tracking algorithm and start a visual servoing routine with gesture commands. Our decision of using the linkage between the user and the robot arm for coarse motions has been validated by our experiments. Teleoperated motions, while fast, are jittery and not very precise. For small movements and precise location, visual servoing is a good complement.

References

[1] Pratt, Gill A. Robot to the rescue. Bulletin of the Atomic Scientists 2014.

[2] Kofman, Jonathan and Wu, Xianghai and Luu, Timothy J and Verma, Siddharth. Teleoperation of a robot manipulator using a vision-based human-robot interface.IEEE Transactions on Industrial Electronics 2005.

[3] Chang, Sokho and Kim, Jungtae and Kim, Insup and Borm, Jin Hwan and Lee, Chongwon and Park, Jong Oh. KIST teleoperation system for humanoid robot.Intelligent Robots and Systems, 1999.

[4] Leeper, Adam Eric and Hsiao, Kaijen and Ciocarlie, Matei and Takayama, Leila and Gossow, David.Strategies for human-in-the-loop robotic Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction 2012.

[5] E. A. Suma, B. Lange, A. Rizzo, D. Krum, M. Bolas, FAAST: The flexible action and articulated Skeleton Toolkit. IEEE Virtual Reality 2011.

[6] H. Suay, S. Chernova. Humanoid Robot Control Using Depth Camera HRI 2011.

[7] R. M. Taylor, T.C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. VRPN: a device-independent, network-transparent VR peripheral system. In ACM Virtual Reality Software and Technology, pages 55-61, 2001.

[8] M. Quiqley, K. Conley, B. Gerkey, J. Faust, T.B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, ROS: an open-source Robot Operating System, in ICRA workshop on OpenSource Software, 2009.

[9] S. Goto, T. Naka and Y. Matsuda, Teleoperation system of robot combined with remote control and visual servo control, SICE Annual Conference 2010.

[10] K. Khoshelham and S. Elberink, Accuracy and resolution of Kinect Depth Data for Indoor Mapping Applications, Sensors 2012

[11] R. Igorevich, E. Ismoilovich, D. Min, Behavioral Synchronization of Human and Humanoid Robot,8th international Conference on Ubiquitous Robots and Ambient Intelligence (URAI2011)

[12] Peter F. Hokayem, Mark W. Spong, Bilateral teleoperation: An historical survey, Automatica, Volume 42, Issue 12, December 2006, Pages 2035-2057

[13] Matthew Marshall, James Matthews, Ai-Ping Hu, Gary McMurray Harvey Lipkin Uncalibrated Visual Servoing for Intuitive Human Guidance of Robots Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA) 2012, To appear

[14] T.B. Sheridan Space teleoperation through time delay, Review and prognosis IEEE Transactions on Robotics and Automation, 9,5, 1993, pp. 592-606.

[15] A. Rachmielowski, N. Birkbeck, M. Jagersand, Performance Evaluation of Monocular Predictive Display , Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA) 2010 p5309 - 5314.

[16] Niemeyer, G., Slotine, J.J.E.(1997b). Using wave variables for system analysis and robot control . In Proceedings of the IEEE international conference on robotics and automation (Vol. 3, pp.1619-1625),Albuquerque, NM,USA.

[17] S. Hutchinson and G. Hager and P. Corke. A tutorial on visual servo control. IEEE Trans. on Robotics and Automation, 1996.

[18] Chaumette, F. and Hutchinson, S. Visual servo control, Part II: Advanced approaches. IEEE Robotics and Automation Magazine, 2007.

[19] A.-M. Farahmand and A. Shademan and M. Jgersand. Global visual-motor estimation for uncalibrated visual servoing. Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.2007.

[20] J. A. Peipmeier and G. V. McMurray and H. Lipkin. Uncalibrated dynamic visual servoing. IEEE Trans. Robot. Automat.2004.

[21] A. Shademan and A.-M. Farahmand and M. Jgersand. Proc. IEEE Int. Conf. Robot. Automat.2010.

[22] H. Sutanto and R. Sharma and V. Varma. The role of exploratory movement in visual servoing without calibration. Robotics and Autonomous Systems. 1998.

[23] Jagersand, M. and Fuentes, O. and Nelson, R.Experimental evaluation of uncalibrated visual servoing for precision manipulation. IEEE International Conference on Robotics and Automation.1997

[24] K. Hosoda and M. Asada. Versatile visual servoing without knowledge of true Jacobian. 1994.

[25] G. Bradsky. Computer Vision Face Tracking For Use in a Perceptual User Interface.Intel Technology Journal Q2 1998.