2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

First-Person Tele-Operation of a Humanoid Robot*

Lars Fritsche[†,1] and Felix Unverzagt[†,1] and Jan Peters[1,2] and Roberto Calandra[1]

***Abstract*— Remote control of robots is often necessary to complete complex unstructured tasks in environments that are inaccessible (e.g. dangerous) for humans. Tele-operation of humanoid robots is often performed trough motion tracking to reduce the complexity deriving from manually controlling a high number of DOF. However, most commercial motion tracking apparatus are expensive and often uncomfortable. Moreover, a limitation of this approach is the need to maintain visual contact with the operated robot, or to employ a second human operator to independently maneuver a camera. As a result, even performing simple tasks heavily depends on the skill and synchronization of the two operators. To alleviate this problem we propose to use augmented-reality to provide the operator with first-person vision and a natural interface to directly control the camera, and at the same time the robot. By integrating recent off-the-shelf technologies, we provide an affordable and intuitive environment composed of Microsoft Kinect, Oculus Rift and haptic *SensorGlove* to tele-operate in first-person humanoid robots. We demonstrate on the humanoid robot *iCub* that this set-up allows to quickly and naturally accomplish complex tasks.**

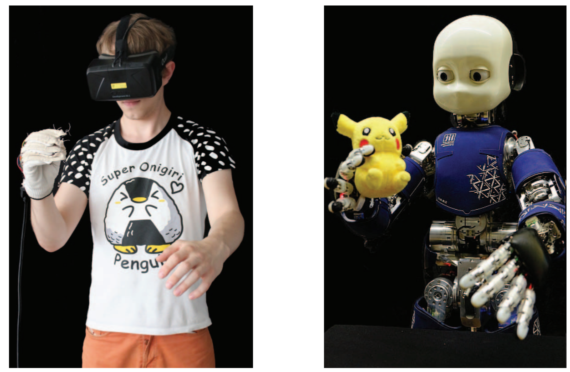


Fig. 1: First-person tele-operation of the humanoid robot *iCubDarmstadt01*. The operator *(on the left)* see trough the eyes of the robot and perceive the haptic feedback from the robot's hand. At the same time, the movements of the operator's upper body (head, torso, arms and hands) are mirrored to the robot *(on the right)* allowing him to interact with the environment.

I. Introduction

Humanoid robots are appealing as they offer the possibility to work in environments that were originally created by humans for humans. However, it is often already challenging and time-consuming to design controllers which perform tasks considered basic for humans, such as turning a valve or assemble a table. Nonetheless, there are many circumstances in which it is crucial to deploy robots, within a limited time-frame, that are capable of performing such tasks in unstructured environments e.g., in presence of natural disasters. For these circumstances, the use of tele-operation is both logical and beneficial as it allows a human operator to directly control the robot. The advantages are the faster deployment (by side-stepping the design of an appropriate task-dependent controller) and the adaptive decision-making capabilities of the human operator. As a result, tele-operation provides the possibility of performing complex tasks in environments otherwise inaccessible or deemed too dangerous for humans. Some examples could be hazardous environments like epidemic areas (e.g., Ebola [1], [2]), damaged nuclear plants (e.g., Fukushima) or insecure urban structures, but also surgical applications [3], space and underwater exploration. Additionally, tele-operation can be used to teach the robot by demonstration to perform tasks in a similar manner to Kinesthetic teaching but without introducing external forces that could change the dynamics of the robot. One key challenge for the tele-operation of humanoid robots is the interface to/from the human operator. Controlling a full humanoid robot with a high number of degrees of freedom (DOF) can be a challenging task. Instead of controlling each single joint separately (e.g., through joystick), motion tracking allows for a natural and effective interface to map human motion directly to robot behaviour through simultaneous control of a high number of DOFs. A popular choice to reduce the cost of commercial motion tracking devices is the use of Microsoft Kinect. In [4] a tele-operation framework using the Kinect for collision-avoidance control of a NAO robot is introduced. The NAO and the Kinect are also used in [5] to record movement and reproduce it in order to support exercise therapy. Another tele-operation project is [6] which uses a full body motion tracking suit and the NAO for tele-operation tasks. In [7] it is introduced an approach which uses a full body motion tracking suit for tele-operation while using machine learning to train neural networks to map sensor data to joint space.

A limitation of these works is the need to maintain direct visual contact with the task space where the robot is operating. This is also true when we tele-operate the hands of a robot, for example in [8] where sensor gloves are used to control the hand of the *iCub* robot. This limitation can be overcome by providing the operator with a first-person-view [9]. The use of wearable virtual-reality devices

*The research leading to these results has received funding from the EC's Seventh Framework Programme (FP7/2007–2013) under grant agreement #600716 (CoDyCo).

† Equally contributing authors
[1] Intelligent Autonomous Systems, TU Darmstadt, Germany calandra@ias.tu-darmstadt.de
[2] Max Planck Institute for Intelligent Systems, Tübingen, Germany mail@jan-peters.net

allows a seamless integration with the motion capture system and additionally provides a natural way of controlling the view of the operator. A broader description of possible tele-control with tele-presence techniques is given by [10] and [11] implements a tele-presence mechanism to connect an operator with a drone.

In this paper, we propose an approach that integrates the strengths of these previous works by providing a first-person tele-operation framework for humanoid robots. Through a real-time tele-operation system based on Kinect, Oculus Rift (OR) and *SensorGlove* [12] we provide an intuitive and affordable interface for the control of humanoid robots. The operator sees through the eyes of the robot and perceives haptic feedback measured from the tactile sensors on the hands of the robot. Furthermore, whole-body movements performed from the operator are tracked and mirrored onto the robot.

II. Experimental Setting

In this section, the hardware setting used for the tele-operation is described: Microsoft Kinect, Oculus Rift, *SensorGlove* and the humanoid robot *iCub*.

A. Microsoft Kinect Version 2

To collect motion data of the body of the human operator we use the Microsoft Kinect v2 [13]. It combines a camera and a depth sensor to retrieve 3D-skeleton data from bodies. The Kinect v2 improves the Kinect v1 in terms of resolution and accuracy [14]. The advantages of this device, compared to full body tracking suites, are the low price and its use without the need of special cameras and markers. Besides, the computations needed to approximate the position of the human skeleton are completely outsourced to the device and do not have be performed on the main system. Disadvantages that come along with using this technology are: 1) The tracking with the Kinect it is not as accurate as body tracking suits since it relies on the field of vision which may get occluded (e.g., by other parts of the body); 2) The update frequency for the skeleton tracking is limited to 30Hz; 3) Even under optimal conditions, the raw data can be shown to be noisy and should not be directly used for high precision tasks. To collect the data from the Kinect, we use the official Microsoft Kinect SDK v2.0 which manages the execution of the device and provides an interface for retrieving the Cartesian coordinates of 25 identified joints. The SDK is theoretically able to handle multiple Kinect devices concurrently and by combining their field of vision improves the tracking of the operator in presence of occlusions[1].

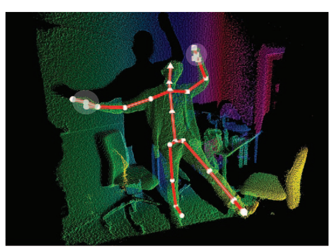


Fig. 2: Example of skeleton tracking with the Kinect

[1] We could not integrate the use of multiple Kinects in our work due to special hardware requirements for activating this feature.

B. Oculus Rift Development Kit 2

The Oculus Rift DK 2 is a head-mounted virtual reality display currently in beta testing phase. It consists of a high resolution display, split vertically (one half for each eye) and a gyroscope to accurately track the head movement with a frequency of 1 kHz which is needed to avoid motion sickness. The control of the OR is performed via the official Oculus Rift SDK which allows to access the head orientation data as well as sending images to the display.

C. SensorGlove

In the context of tele-operation it is often essential to support accurate grasping motions. In this paper we use haptic feedback sensor gloves developed at TU Darmstadt [12]. These *SensorGlove*s are based on the design introduced in [15] and allow to track finger motions and provide haptic feedback to the operator. Via a serial communication interface, we can access the flex sensors that measure the bending of each fingers at around 350Hz. Furthermore, we can control the vibration motors to induce haptic feedback on the fingertips of the operator according to the pressure measured on the fingertips of the robot.

D. The Humanoid Robot iCub

The *iCub* is a humanoid robot built at the Instituto Italiano di Tecnologia (IIT) with the approximate size of a 6 year old child: 104 cm of height and 24 kg of weight [16]. The *iCub* possesses 53 degrees of freedom (DOFs) of which, we actively control 30 during the tele-operation: 3 DOFs for the torso, 5 DOFs in the head, 4 DOFs in each arm and 7 DOFs for each hand. Additionally, the robot is provided with tactile sensors on the fingertips to measure the pressure, and two cameras that provide images at the maximum resolution of 640 x 480 pixels for each eye. These cameras are parts of the DOF of the head and can therefore be moved. The *iCub* uses YARP [17] as a way to define input and output ports for its control. The use of YARP allows the *iCub* to be modular and easily extensible as each input and output port can be activated/deactivated separately. Furthermore, it enables modules to run concurrently and interchange messages in a non-blocking way.

III. Control Architecture

The structure of the implemented software architecture is depicted in Figure 3, and consists of four separate modules: Kinect adapter, Oculus adapter, *SensorGlove* adapter, and the *iCub* controller. The first three modules manage the data collection of their respective devices: Kinect, OR and *SensorGlove*. For the OR, its adapter also takes care of processing images from the stereo cameras of the *iCub* and displaying them on the OR's screen. However, the use of both *iCub*'s cameras requires a mechanical calibration procedure. For robots that do not offer stereo cameras or for which calibration is impractical, an alternative approach is to project the images from one single camera to both eyes of the OR. For the *SensorGlove*, its adapter also takes care of controlling the vibration motors on the fingertips, based on

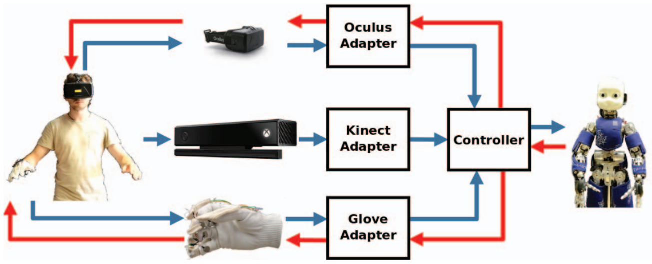


Fig. 3: Modular Software Architecture - Consisting of three modules to read out data from Kinect, Oculus Rift and Sensor Gloves and another module to control the robot. The blue lines indicate position data measured on the operator side and resulting into motion of the *iCub* robot. The red lines indicate visual (from the eye cameras) and tactile (from the robot skin on the fingertips) data collected on the robot and provided as feedback to the operator.

the pressure registered by the robot fingertips tactile sensors. Every module implements a YARP-interface used to send and receive all data into and from the YARP-network. This data is then used by the fourth module, the controller, to derive actual motions control for the *iCub*. For the Kinect, the controller processes the skeleton data of the operator to generate the corresponding joint angles to control the upper-body of the robot. In case of the OR and *SensorGlove*, the controller derives the position of the operators head and the angle of its fingers. Crucially, the controller implements filtering of the data to increase the stability of the derived trajectories and adds safety limitations to the *iCub* joint configuration to avoid self-collisions and to decrease the risk of damaging the motors. Finally, the control signals generated by the controller are sent to the corresponding YARP-Ports of the *iCub* which are independently defined for the arms, legs, torso and head. The controller is also tasked with retrieving data from the *iCub*, making them available to the adapters that require them and logging the current state of all the devices (including the *iCub*). This object-oriented architecture clearly modularizes the components of the system, which is critical to port the architecture to other robots that are based on YARP. Such porting can be done by simply exchanging the controller while keeping the other modules untouched. At the same time, additional motion tracking or feedback devices can be integrated easily by creating a new adapter and extending the controller. Finally, each module can be executed on a separate remote machine. This feature is especially important for the use of both OR and Kinect, as they currently have to be operated on a computer using Windows as operating system. The complete code developed is available open source at `https://github.com/robertocalandra/firstperson-teleoperation`.

In the next subsections we give a detailed description of the implemented safety routines and data filtering.

A. Safety Routines

Multiple safety routines had to be implemented to guarantee a robust tele-operation in different conditions and even in presence of non-expert operators. A first safety routine had to be implemented because the *iCub* itself can not approach all of its workspace safely: reaching the joint limits occasionally lead to heavily jerking behavior that endangers the whole robot. Hence, we implement additional thresholds for each joint that further limit the workspace. These prerequisites are already made under the assumption that the operator that controls the robot can be seen clearly by the camera and that no other people are in the line of sight. When multiple people are visible in the scene by the Kinect, it is not trivial to track a specific one, e.g. when somebody passes between the camera and the operator, the Kinect could lose track of the operator. After obscuring the operator he would be re-detected, but as a completely new user. This can be avoided by checking the number of people that are tracked by the Kinect. In case that there are more than one, it is safer to cancel all actions and to wait until only one person is left. However, this procedure introduces a new problem: while continuing to send commands to the robot, it might be that both the operator and the robot are not in the same pose anymore. If the difference in the operators posture and that of the robot is very high, and depending on the resulting velocities, the robot might approach his new posture too fast. A similar problem arises for the OR when the operator decides to interrupt the head-tracking manually. As a result, we implemented safety routine that consider these issues. The low-level control of the robot is performed using the so called “direct control”, which means that no trajectory generator but only a simple linear interpolation is used to reach the desired point with maximum velocity. This mode is suitable for small position changes but may be very dangerous in case of large changes, e.g. in case of small update rates. For this reason we introduced another safety measure to guarantee a maximum step size for the command signal such that the robot can not exceed its own bounds by force. To summarize, there are various scenarios that have to be taken into account while tele-operating the *iCub* as it is not safe by construction.

In our experience, we found that it is important to validate the tele-operation and safety routines in a safe environment. For this purpose, and specifically for the *iCub*, we used the provided full simulation environment based on the Gazebo simulator [18]. Nonetheless, it is important to notice that the *iCub* often acts more stable within the simulation than in the real world, e.g., due to inaccuracies in the dynamics model.

B. Filtering of the Control Signals

For all the three devices used for tele-operation we implemented filters to smooth the control signals. The data retrieved by the Kinect are noisy by construction and would lead to very jerky movements if applied directly to the robot. In case of the OR, the accuracy is very high, nonetheless the sampled data generates abrupt direction changes on a very low scale that could endanger the motors. Hence, the data from the OR has to be smoothed, too.

As tool of choice for our filters we used Butterworth filters. Since the noise properties of the devices are inhomogeneous it was necessary to design two individual filters: one filter was designed for the OR and *SensorGlove*, and one for the Kinect. Generally, the parameters that define a Butterworth filter are maximum frequency, cutoff frequency and order. The maximum frequency is the sample rate which in this case is limited to 100Hz for all three devices[2]. For the Kinect filter the cutoff frequency was set to 1.5Hz while for the OR and *SensorGlove* filter it is 5Hz. The Kinect filter has been chosen to be of eight order which increases the response delay but also highly stabilizes the signal. This rather high delay is a necessary trade-off to guarantee a certain amount of safety. The filter for both *SensorGlove* and OR are of fourth order since the unfiltered signal is more stable than the Kinect. The advantage of the lesser order and lower cutoff frequency is a decreased delay and a smoother signal, which benefits the immersion of the operator when turning his head or closing his hand. In Section V-A we evaluate and analyze the performance of the designed filters.

IV. Operator-Robot Correspondence

In this section, we show how poses recorded from the operator can be converted to robot configuration.

A. Mapping of the Arm Joints

The *iCub* has 7 DOF in its arm from which we control the first 4 (pitch, roll, yaw of the shoulder and yaw of the elbow). The remaining 3 DOF are not actively controlled since they are used for the orientation of the hand, and our setting does not currently provide reliable data to control them. To retrieve the controls for the other DOF, we draw the positions of the operators shoulders, elbow and wrist from the Kinect. These positions are represented as points in 3-D space. From the position of the shoulder and elbow, we calculate the pitch and roll of the *iCub*s shoulder using the kinematics of the robot. By drawing a vector from the shoulder to the elbow and another from the elbow to the wrist, we can calculate the yaw of the elbow as the angle between those vectors. Given the shoulders pitch, roll and the elbows yaw, the shoulders yaw can now be calculated with kinematics from the wrists position to the shoulder. We also evaluated other mapping techniques, but this one resulted to be the most robust in our experiments.

B. Mapping of the Torso Joints

The torso of the *iCub* has 3 joints: pitch, roll and yaw. All calculations for the torso are made under the assumption, that in case the operator is standing upright, his spine is aligned with the gravity vector. Furthermore, we assume that the Kinects sight is aligned parallel to the ground. Under these assumptions, the data to control the robot can be derived from the operators 3-D positions of his spine, hip and shoulders that are drawn from the Kinect.

C. Mapping of the Head Joints

The head of the *iCub* has 6 DoFs: roll, pitch and yaw of the neck, and vertical, horizontal and squinting of the eyes[3]. The data to control the head are retrieved by the OR which measures the 3 angles of the neck. Since the OR can not track the eye movement and the *iCub* has a natural limit for pitch and yaw, we use the commands that would exceed those bounds to move the eyes in the corresponding direction. As a result, we increase the line of sight on the *iCub* from $\pm 35°$ (by solely tracking the head motion) to $\pm 65°$ (additionally using the eyes). We evaluate the tracking performance using the additional eyes movement in Section V-B.

D. Mapping of the Hand Joints

We control 7 DoFs in each hand of the *iCub*: the thump, index and middle finger are controlled each by 2 DoFs, while ring and small finger are controlled simultaneously by a a single DoF. The bending of the operators fingers is measured by the *SensorGlove*. Additionally, the *SensorGlove* returns haptic feedback in form of vibrations on the operators fingertips according to the applied pressure on the fingertips of the *iCub*. The intensity of the vibration is represented as a value in the range of [0, 255] which is mapped to the range of pressure from the *iCub*s tactile sensors on its fingertips. Each fingertip has 12 tactile sensors of which the maximal value is used as the vibration intensity.

V. Experimental Results

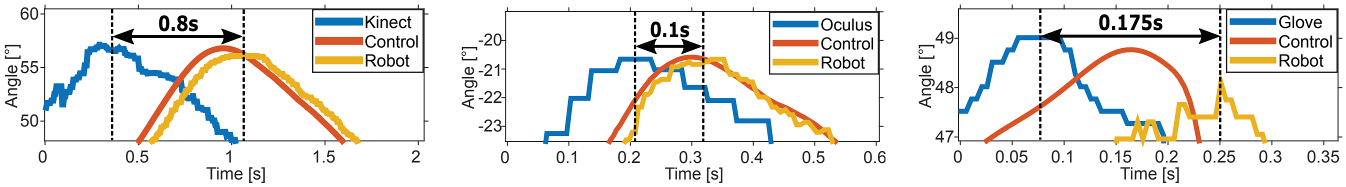
In this section, we experimentally evaluate our proposed setting. We first evaluate the delay between operator and robot. Following, we demonstrate that the robot is capable of reproducing simple movements performed by the operator. Finally, we demonstrate our setting on a pick-and-place task.

A. Filter and Delay Evaluation

A challenge in designing tele-operation systems is to provide a low delay between the motion of the operator and the induced motion on the robot. Nonetheless, due to unreliable motion tracking data, safety routines have to be integrated such that a trade-off between safety and delay is found. In Figure 4 this trade-off can be seen for the Kinect, OR and *SensorGlove*. The delay introduced by the Butterworth filter to the data derived by the Kinect is approx. 600ms while the *iCub* needs another 200ms to follow-up. For the OR and *SensorGlove* the delay of the filter to the data is approx. 100ms for both. While the *iCub*'s head is able to follow-up the control signal without any noticeable delay, its fingers need 75ms to reach the desired position. The delay of the Kinect is rather high but due to instabilities this is a necessary compromise to avoid jerky movements of the upper body which can potentially damage the *iCub*. For the OR, the delay is chosen rather low since the raw data is very accurate. Such low delay is essential during tele-operation since a high delay in the visual perception can be highly disturbing for the operator. Additionally to the discussed

[2] The Kinect has a refresh frequency of 30Hz which is boosted to 100Hz.

[3] Squinting is not controlled in our approach.



(a) Time delay for the Kinect (b) Time delay for the Oculus (c) Time delay for the *SensorGlove*

Fig. 4: Examples of peak-to-peak delay for each device, from measurement of the operator motion to execution of the motion on the *iCub*. The blue curve shows the data measured from the device, the red curve shows the control signal applied to the *iCub* after filtering and safety checks, and the orange curve is the measured motion of the robot. The delay for the Kinect is the highest due to the cautious filtering used, but still within acceptable limits to effectively perform tele-operation tasks.

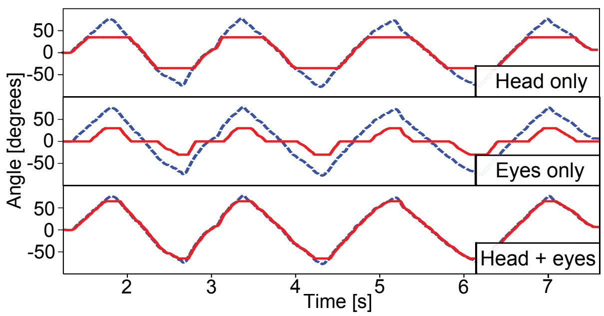


Fig. 5: Decomposition of the operators head motion into head and eye motion on the *iCub*. To follow the desired trajectory of the operator's head (blue dashed), the head is moved first. Once the *iCub*'s head reach its limits, the eyes start moving, allowing a movement of 65 degrees in each direction.

delay for the head, there is another one for the vision caused by the communication of the images through the network and theirs rendering. We do not measure this delay but empirical results suggest that the impact is rather small in comparison to the control delay and can be disregarded. Overall, the most important result was that so far none of our operators experienced motion sickness despite the head motion delay.

B. Mimic Task

As can be seen in Figure 6, the robot is able to mimic the head motion performed by the operator. The operator sees through the eyes of the robot and is able to control its behavior within ego perspective. As the joint limits of the head are reached, it can be seen that reaching this limits leads to a movement of the eyes in the corresponding direction. This increases the field of view which the robot can access by (+/-) 30 degree without the need to turn around as shown in Figure 5. Figure 7 shows a combination of body and head movement, using the Kinect and OR, respectively. The images depict how the robot mimics the movement of the upper body, including both arms and torso.

C. Pick-and-place Task

Figure 8 shows the robots ability to interact with objects. The experimental arrangement consisted of the full first-person tele-operation setting (i.e. Kinect, OR and *SensorGlove*) that is used to pick-up and return an object from and to another interacting person. A crucial part of such task was to safely interact with a human. This goal was achieved by the introduced filter technique which stabilized the control signals in such a way that a safe interaction could be realized successfully. However, the task itself is also challenging for other reasons: 1) The estimation of depth, which arises from the different limp lengths of the *iCub* and the missing stereo-vision[4]. 2) The introduced filter delay that the operator has to get used to.

The results of the experiments showed that although the operator dropped the object several times in the beginning, he was able to rapidly adapt to the different dimensions of the *iCub*. After a couple of tries, the operator was able to frequently pick-up and return the object at different locations without dropping it[5]. In our implementation, the *iCub* does not control its wrist nor provide stereo-vision for the operator. From our experience, the addition of these features seems an important and promising way to increase the immersion and efficiency of the tele-operation.

VI. Discussion & Conclusions

In many environments where the human presence is either dangerous or impossible, tele-operation is an effective way to execute complex tasks. As today, the tele-operation of humanoids robots remains a challenging task which heavily relies on the skills of the operators. In this paper we present an affordable approach to first-person tele-operation using the Microsoft Kinect, Oculus Rift and *SensorGlove*. This approach introduces augmented reality on the humanoid robot *iCub* to achieve a more natural and efficient tele-operation. We thoroughly describe the proposed setting, with a special emphasis on the safety measure adopted and the resulting tele-operation delays. We demonstrate this approach on the humanoid *iCub* on a pick-and-place task and show that it allows to complete complex tasks with ease. Additionally, the proposed setting can be used to easily record human motion data and produce demonstrations to be used in imitation learning, without introducing external forces.

In future work, the existing setting will be used for psychological studies of effects deriving from the use of first-person tele-operation and body schema adaptation. Another possible future extension is the introduction of supportive features for semi-automatic operation, such as in grasping.

[4]Hardware calibration of the eyes of the *iCub* is rather challenging
[5]A video is available at `https://youtu.be/gtkrPhcYhyI`

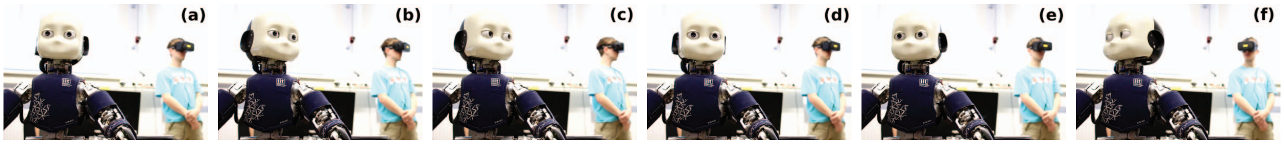


Fig. 6: Example of tele-operation of the head only (neck + eyes)

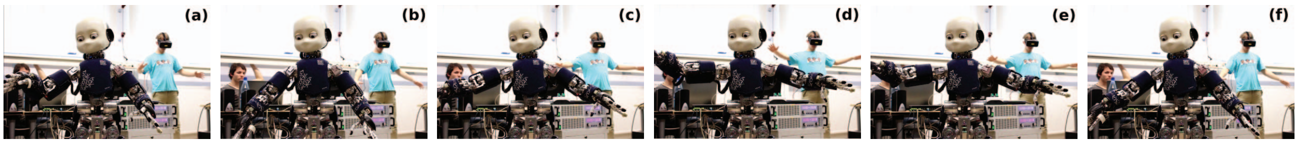


Fig. 7: Example of tele-operation of the upper body and head

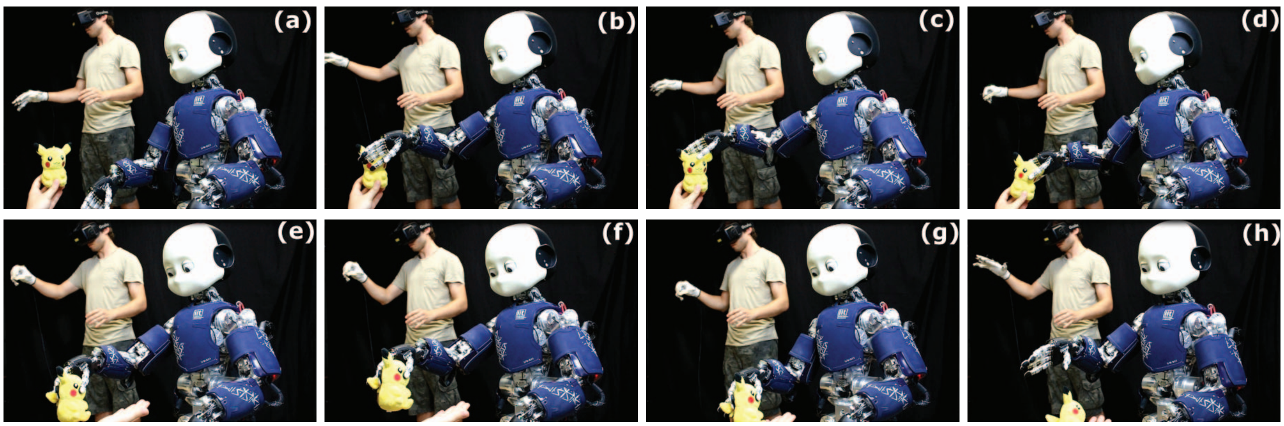


Fig. 8: Grasping task: The experimental setting combines the use of Kinect, Oculus Rift and *SensorGlove* to tele-operate the *iCub*. The performed task was to safely pick-up and return an object to an interacting person.

ACKNOWLEDGMENT

We thank Philipp Beckerle and Marius Schmidt from the Institute for Mechatronic Systems at TU Darmstadt for the development of the *SensorGlove*, and Elmar Rueckert for helping to develop its interface.

REFERENCES

[1] "Exploring opportunities for robotics to aid in disease outbreaks." [Online]. Available: http://www.whitehouse.gov/blog/2014/11/17/exploring-opportunities-robotics-aid-disease-outbreaks

[2] "Ebola grand challenge." [Online]. Available: http://www.ebolagrandchallenge.net/the-problem-detail/

[3] G. H. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surgical Endoscopy and Other Interventional Techniques*, vol. 16, no. 10, pp. 1389–1402, 2002.

[4] S. Filiatrault and A.-M. Cretu, "Human arm motion imitation by a humanoid robot," in *IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, Oct 2014, pp. 31–36.

[5] F. W. Sven Franz, Ralph Nolte-Holube, "NAFOME: NAO Follows Me - tracking, reproduction and simulation of human motion," Jade University of Applied Sciences, Germany.

[6] J. Koenemann, F. Burget, M. Bennewitz, F. Burget, M. Cenciarini, B. Meier, H. Bast, M. Bennewitz, W. Burgard, C. Maurer, *et al.*, "Real-time imitation of human whole-body motions by humanoids." *Autonomous Robots*, 2013.

[7] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," in *Proc. Australasian Conference on Robotics and Automation*, 2012.

[8] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang, "Precision grasp synergies for dexterous robotic hands," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2013, pp. 62–67.

[9] C. Pittman and J. J. LaViola, Jr., "Exploring head tracked head mounted displays for first person robot teleoperation," in *International Conference on Intelligent User Interfaces (IUI)*, 2014, pp. 323–328.

[10] L. Almeida, B. Patrao, P. Menezes, and J. Dias, "Be the robot: Human embodiment in tele-operation driving tasks," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014, pp. 477–482.

[11] K. Higuchi, K. Fujii, and J. Rekimoto, "Flying head: A head-synchronization mechanism for flying telepresence," in *International Conference on Artificial Reality and Telexistence (ICAT)*, 2013.

[12] E. Rueckert, R. Lioutikov, R. Calandra, M. Schmidt, P. Beckerle, and J. Peters, "Low-cost sensor glove with force feedback for learning from demonstrations using probabilistic trajectory representations," ICRA2015 Workshop on Tactile & force sensing for autonomous, compliant, intelligent robots, 2015.

[13] "Official kinect website," http://www.microsoft.com/en-us/kinectforwindows/, accessed: 2015-07-04.

[14] H. Gonzalez-Jorge, P. Rodríguez-Gonzálvez, J. Martínez-Sánchez, D. González-Aguilera, P. Arias, M. Gesto, and L. Díaz-Vilariño, "Metrological comparison between Kinect I and Kinect II sensors," *Measurement*, vol. 70, pp. 21–26, 2015.

[15] A. De Beir, E. Caspar, F. Yernaux, P. M. Da Saldanha da Gama, B. Vanderborght, and A. Cleermans, "Developing new frontiers in the rubber hand illusion: Design of an open source robotic hand to better understand prosthetics," in *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014, pp. 905–910.

[16] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.

[17] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal on Advanced Robotics Systems*, 2006.

[18] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2149–2154.