2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Exploiting eye-hand coordination: a novel approach to remote manipulation

Lukas Twardon, Andrea Finke and Helge Ritter
EXC Cognitive Interaction Technology - CITEC
Bielefeld University, Germany
{ltwardon, afinke, helge}@techfak.uni-bielefeld.de

Abstract— **Eye movements play an essential role in planning and executing manual actions. Eye-hand coordination is a natural human skill. We exploit this skill for an intuitive remote manipulation system that allows even non-expert users to operate a robot safely without prior experience. Specifically, we propose a visio-haptic approach to controlling a 7-DOF robotic arm. Our system is fully mobile, allowing for unconstraint operation in any environment. An eyetracker captures the operator's gaze. The end effector or particular joints are selected by simply fixating the to-be-controlled segment. A sensor-equipped tangible object provides a haptic interface between the operator's hand and the focused part of the robotic arm. The system features two operation modes, direct joint rotation and 3d end effector control in a global cartesian frame. We evaluated the system in a proof-of-concept study with untrained users. The participants safely operated the robot and accomplished an obstacle avoidance task. For this purpose, they used both operation modes.**

I. Introduction

Remote manipulators are used in various fields like manufacturing, mining or scientific research. They are preferably applied in environments that are too complex for today's autonomous robots. In these situations, a human operator usually directly controls the manipulator. Therefore, the human-robot interface ought to be intuitive and adapted to the user's skills. Researchers have proposed several operation approaches beyond classical keyboard or joystick control. Hand or arm tracking permits online mapping of human body joints to the joints of a robotic hand [1] or a mechanical arm [2]. Robots may be operated through gestures [3] and even thoughts [4]. While all these interfaces try to take advantage of one or another natural human ability, none of them take into account the interplay of eye gaze and manual actions.

Humans rely on a well-organized interaction between vision and haptics in many everyday tasks. The capability to perform hand and eye movements simultaneously in order to reach a common goal is referred to as eye-hand coordination. Eye gaze supports hand motion planning by identifying landmarks that are critical for the completion of a task [5]. Most manual tasks involve a number of subgoals like touching, grasping or moving an object. Here, gaze is directed to the target object for a certain period of time until shortly after the subgoal was reached [6].

We argue that gaze reveals the operator's intentions during remote manipulation. Operating a robot is a task that consists of multiple subtasks and as such generates eye movements that give us an indication of the user's motion planning. Gaze can be measured with an eyetracker. Eyetracking is not only a popular research method in psychology and neuroscience, but has also been used online in order to implement input devices for handicapped people [7]. Gaze-based human-machine interfaces are now becoming more and more interesting for a wider range of user groups as eyetrackers are getting smaller, cheaper and more portable.

We have developed a system that combines gaze-control with haptic remote manipulation, thus allowing the user to intuitively operate a robotic arm. Our human-robot interface consists of two basic input components, namely an eyetracker and a tangible object equipped with tactile and inertial sensors. During operation, the user is free to walk around the robot as the interface is mobile and fully wireless. He should feel as if he could touch and grab the robot, but without the risk of real physical contact.

II. Related work

Several robot control devices based on accelerometers or inertial measurement units have been proposed recently. Uribe et al. [8] used a Wiimote game controller to teleoperate a holonomic robot and found it to be more intuitive than other haptic user interfaces. Wrist rotations turned out to be a very natural way to control a robot. Neto et al. [9] operated an industrial robotic arm using predefined gestures which were captured by accelerometers. Systems trying to estimate position from acceleration always suffer from errors accumulating over time. Smith and Christensen [10] cope with this issue by assuming a parameterized human motion model.

Eyetracking has been considered to be helpful in teleoperation. Remote cameras can be gaze-controlled in order to solve the hands-busy problem and to reduce the perceptual load [11]. Atienza and Zelinsky [12] used 3d gaze tracking to detect objects fixated by the user. A robotic arm picked up these objects of interest and handed them over to the user. Contrary to this task-level approach, Latif et al. [13] presented an interface for immediate gaze-based motion control. For example, looking at the right part of a live video image triggered a turn right movement of a mobile robot.

While both haptic and gaze-control have been well studied, interfaces that combine these two modalities are less common. However, a few multimodal approaches exist. Latif et al. [13] extended their interface by a foot pedal accelerator because the exclusive use of eye gaze led to an increase in the task's workload. Úbeda et al. [14] proposed a video-based teleoperation system that allows the user to move a robot end effector in a two-dimensional plane with a combination of eye and hand movements. To the best of our knowledge, the prototype we have developed is the first completely mobile remote manipulation interface to integrate gaze and touch.

III. Visio-haptic remote manipulation

Our robot control interface exploits the fact that humans are usually very good at coordinating eye and hand movements. Gaze determines *what* is to be handled while manual action specifies *how* something is handled. In our scenario, the user remote controls a redundant robotic arm. The user wears a head-mounted eyetracker that tracks his gaze during the manipulation task. Also, he holds a sensor-equipped object (called *iObject*) in his hand. The robot can be operated in two different control modes which are shown in Figures 1(a) and 1(b). The workflow of our system is as follows:

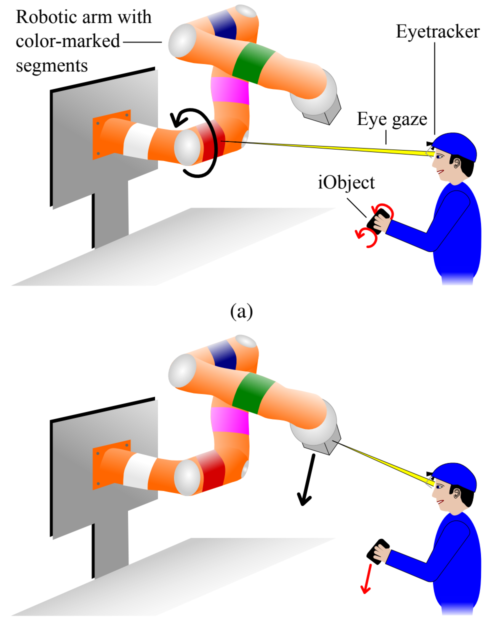
- **Joint control:** Concurrently looking at a segment of the robotic arm and gently squeezing the *iObject*, the user can rotate the respective joint by rotating the object in his hand. As soon as the object has been pressed, the user is not bound to continuously gaze at the segment. The joint remains selected until the user releases the object.
- **End effector control:** It is also possible to move the robot end effector in the 3d global frame by moving the *iObject*. This operating mode is activated by looking at the end effector and pressing the object at the same time. Afterwards, the user is again free to look around. After two seconds, he must release the object, stop, and grip it again in order to move on. This is due to stability issues with position estimation from acceleration (see section V-E). Hence, 3d end effector control is a step-by-step process.

The end effector control mode is very helpful for roughly positioning the robot since the system supports the operator by computing the inverse kinematics. Direct joint control, in contrast, can be used to cope with singularities and to avoid obstacles. There is no need for explicit mode switches as the system "reads" the user's intentions from his eyes. For example, fixating the end effector automatically triggers a mode switch to end effector control. Considering more complex scenarios, the visio-haptic approach is highly extendable. Quickly selecting from a number of manual actions with gaze is even more reasonable in situations with multiple tools or robots.

IV. System setup

A. Robotic arm

We use the *Kuka Lightweight Robot*, a highly redundant 7-DOF robotic arm. We have marked the arm segments with either colored paper or BCH code fiducial markers (see Figure 5). This makes the segments detectable in the scene camera image from the eyetracker. Also, instead of a real tool or gripper, we have attached a cube with fiducial markers at the end effector. Thus, no sophisticated object recognition algorithm is required for our testing purposes.



(a)

(b)

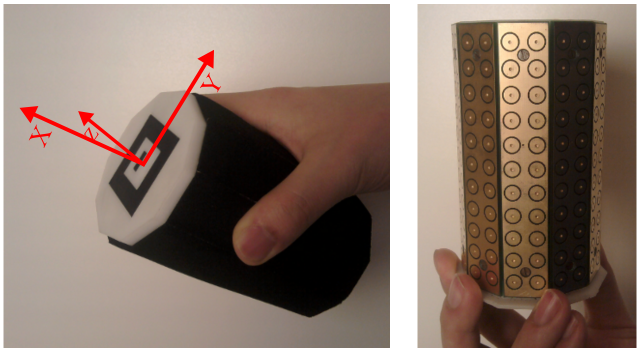
Fig. 1. General functionality of the visio-haptic interface. (a) Direct joint control. (b) 3d end effector control.

B. Eyetracker

We chose the monocular *SMI iView X* eyetracker which is head-mounted and can be used in dynamic environments. The controlling laptop can be carried in a backpack so that the system is fully mobile. The eyetracker is video-based having both an eye- and a scene camera attached to a helmet. The scene camera records the user's field of view. The eye camera tracks the pupil and the corneal reflection in order to determine the gaze direction in real-time. The output of the eyetracking module is a scene video stream plus the current gaze point in scene image coordinates.

C. iObject

iObject (intelligent Object) is a novel tangible interface that has been presented by Kõiva et al. [15]. It has been modelled on the shape and size of a standard beverage can (see Figure 2). Thus, it perfectly fits in an average human hand and is suited for all kinds of manipulation tasks. The object's surface is equipped with ten 2×11 tactile sensor arrays. Furthermore, there is an *Xsens MTx* inertial measurement unit (IMU) integrated inside the *iObject*. *Xsens MTx* measures acceleration, orientation and the magnetic field. All captured data are wirelessly streamed via bluetooth. We utilize tactile data as well as acceleration and orientation data. Orientation is represented as a quaternion which describes the rotation of a vector in the sensor coordinate system to a global coordinate system with respect to local magnetic north (see sections V-D and V-E).



(a) (b)

Fig. 2. The intelligent object *(iObject)*. (a) *iObject* with the local coordinate system of the *Xsens MTx* IMU depicted in red. (b) Tactile sensor arrays placed on the *iObject*'s surface.

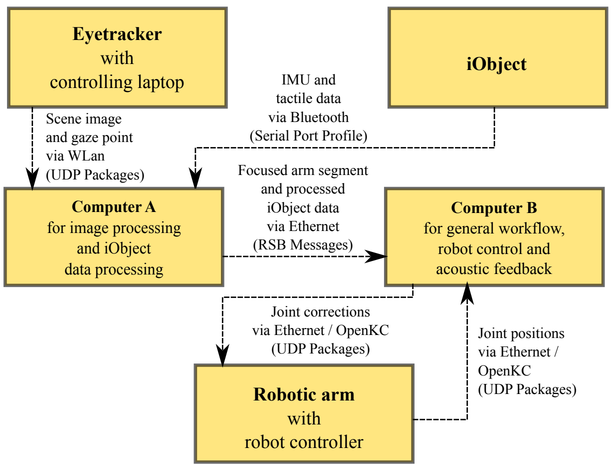


Fig. 3. Communication between the components of the visio-haptic remote manipulation system.

D. Communication

Our framework is a distributed system that consists of five main hardware components. First, we have the input devices, *iObject* and eyetracker. Then, there is the robotic arm which is to be controlled. In between are two computers A and B, one processing the input data, the other controlling the robot and generating acoustic feedback. The input devices must be untethered in order to allow the user to move freely around the robot. Therefore, the *iObject* sends its data via bluetooth while the eyetracker is connected to the stationary computer A over WiFi. Computers A and B communicate in a message-oriented manner using the middleware *RSB* [16]. *OpenKC* [17] encapsulates the UDP package exchange between Computer B and the robot controller. Figure 3 shows an overview of the communication between the components.

V. Methods

A. Robot control

Position and velocity control is performed in joint space. For this purpose, we use *OpenKC* [17], an open source real-time control library for the *Kuka* robotic arm. The library provides a callback function which is executed once at each control step. The function supplies the currently measured joint positions and passes the requested joint corrections back to the robot. The *Kinematics and Dynamics Library (KDL)*[1] is used to solve the inverse kinematics in end effector control mode. We must take into account not only joint angle limits but also 3d workspace limits. Therefore, the forward kinematics is solved for the end effector and for individual segments of the robotic arm.

B. Scene image processing

The eyetracker provides us with a scene video stream reflecting the operator's field of view and with the current gaze point in scene image coordinates (see Figure 4(a)). From this information, we try to determine whether the user looks at a segment or at the end effector of the robotic arm. It is non-trivial to locate and recognize parts of the robot in the scene image.

We implemented a marker-based solution. The segments were marked with either colored paper or BCH code fiducials. Color detection is very sensitive to lighting conditions and background distractions. In our environment, the number of distinguishable colors was limited to four. BCH code marker detection, in contrast, is error-correcting but fails if the fiducials are bent or partially covered. A combination of both approaches gave the best results. We put BCH code fiducials on the first and last segment as well as on the end effector. The other segments were color-marked.

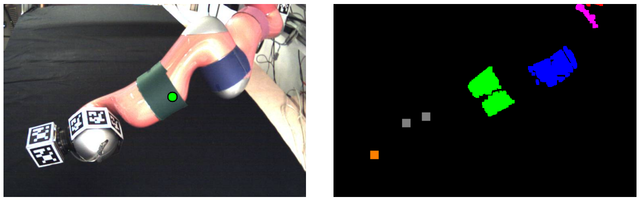
We use the *Image Component Library (ICL)*[2] for the processing of the scene image. Color segmentation is accomplished by accessing a lookup table that maps color values to class labels. The YUV color space is used in order to achieve a certain degree of independence from lighting conditions. We apply an erosion filter followed by a dilation filter on the color segmented scene image so as to remove smaller unconnected areas. If the number of pixels of a specific color class exceeds a predefined threshold, we suppose that we found a color-marked segment of the robotic arm.

ICL implements a BCH code marker detection algorithm that outputs the coordinates of the fiducial markers found in the scene image. The color segmentation and marker detection results are shown in Figure 4(b). We determine the minimum distance between the gaze point and all detected fiducial markers and color regions. If the minimum distance is below a threshold, we assume that the user is looking at the respective segment.

C. iObject grip detection

Both joint control and end effector control are activated by slightly squeezing the *iObject* and deactivated by releasing it. We average over all 120 measured values tac from the tactile sensor arrays placed on the object's surface. Then, we define two thresholds θ_{press} and $\theta_{release}$ with $\theta_{press} > \theta_{release}$. Joint or end effector control is (gaze-dependently) activated, if $\overline{tac} > \theta_{press}$ and deactivated if $\overline{tac} < \theta_{release}$.

[1] `http://www.orocos.org/kdl`
[2] `http://www.iclcv.org`



(a) (b)

Fig. 4. (a) Scene image from the eyetracker's front camera. The gaze point is depicted in green indicating that the image processing algorithm has correctly determined that the user's focus is on the green segment of the robotic arm. (b) Color segmented scene image. Detected fiducial markers are depicted in gray and orange, respectively.

D. iObject rotation tracking

The *iObject* can be thought of as a haptic representation of the focused segment of the robotic arm. Therefore, in direct joint control mode, we directly map a rotation around the object's local x-axis to a joint rotation (in an angle ratio of 4:1). The calculation of the amount of object rotation around the x-axis is described below.

The *Xsens MTx* sensor has been configured to output the object orientation as a quaternion representing the rotation of a vector in the object frame to a north-based global coordinate system. Quaternions form a four-dimensional number system which can be used to describe orientations and rotations. Let q be a unit quaternion, and let v be a vector in $\mathbb{R}^3$. Then, qvq^{-1} is a rotation of v where q^{-1} is the conjugate of q. Quaternion multiplication is a non-commutative operation which results in a quaternion that represents the composition of the input rotations.

In our case, we have two orientation quaternions q_1 and q_2 from two consecutive measurements. We can compute a quaternion q' representing the rotation between both measurements. q' expressed in the object's frame is

$$q' = q_1^{-1} q_2. \tag{1}$$

When we convert q' to its axis-angle representation, $angle(q')$ gives us the angle of rotation. However, we are only interested in the part of the rotation which is around the x-axis. $axis(q')$ is a vector with length one representing the rotation axis. Hence, the requested x-axis part $rot_{iObject}$ of the rotation can be calculated from $angle(q')$ and the x-component of $axis(q')$ as

$$rot_{iObject} = axis_x(q') \cdot angle(q'). \tag{2}$$

The direction of rotation is correct as long as the segment of the robotic arm and the *iObject*'s x-axis roughly point in the same direction. But robot segment and *iObject* are contrarotating when, for example, the segment points downwards and the user holds the object upwards. In these cases, we invert the direction of rotation in order to make the behaviour more intuitive. This is accomplished by multiplying $rot_{iObject}$ by the sign of the scalar product of both axis vectors:

$$rot'_{iObject} = rot_{iObject} \cdot sgn(v_{iObject} \cdot v_{segment}) \tag{3}$$

Here, $v_{iObject}$ is the *iObject*'s x-axis expressed in the global frame. $v_{segment}$ is the selected robot segment represented as a vector which is calculated by solving the forward kinematics. The vector $v_{iObject}$ must be transformed to fit the robot's coordinate system (see section V-E).

E. iObject position tracking

In end effector control mode, *iObject* movement in the 3d global frame is mapped to the corresponding robot end effector movement. We do not use any external references for *iObject* position tracking so as to keep the interface mobile and portable. Position can, in principle, be determined iteratively from acceleration which is called dead reckoning.

For this purpose, the object referenced acceleration vector a from the *Xsens MTx* sensor is transferred to the global coordinate system using the current orientation quaternion q:

$$a' = qaq^{-1} \tag{4}$$

In the global frame, we substract the gravity vector g:

$$a'' = a' - g \tag{5}$$

Then, velocity v is calculated by integrating a'' over time:

$$v = \int a'' dt \tag{6}$$

A second integration step gives us position p:

$$p = \int v dt \tag{7}$$

The dead reckoning approach is known to accumulate errors. According to Walchko and Mason [18] bias and drift are the most devastating sources of error. *Xsens MTx* internally compensates for drift in an appropriate way. However, we still have to cope with the bias issue. Bias is a small offset in acceleration data that grows quadratically over time in terms of position estimation.

Xsens MTx has an acceleration bias of 0.02 m/s^2. Hence, after ten seconds, the position error has already grown to one meter. We handle the bias problem by defining a usage constraint. End effector position control is only allowed for two seconds after having pressed the *iObject*. Then, before he can move on, the operator must release the object and stop. We suppose that there is no hand motion when the user presses the object again and reset velocity to zero. Thus, the accumulated position error for each manual action is limited to about 4 cm. This strategy is sometimes referred to as ZUPT (Zero Velocity Updates) [19].

So far, we have assumed that the global frame of the *iObject* and the robot's coordinate system are identical. Indeed, both are right-handed systems. But the object's global frame is north-based (x-axis points to local magnetic north) while the robot's frame depends on the specific installation.

Also, the *Xsens MTx* compass is not perfectly reliable because of magnetic disturbances. This is why we align the coordinate systems by means of a calibration process. Before using our interface, the operator should position the *iObject* on a table in such a way that its x-axis is parallel to one of the robot's horizontal axes. Then, the rotation quaternion between the measured and expected orientation is determined. This quaternion is used to perform coordinate transformations during runtime.

F. Acoustic feedback

We generate acoustic feedback in two different situations. First, there is speech output ("white", "red", "pink" etc.) when the operator presses the *iObject* and concurrently gaze-selects a color-marked segment or the end effector. Thus, the user is informed about the system's success or failure to determine his intention. Second, an acoustic signal warns the user when he tries to exceed a joint or workspace limit.

VI. User Study

We conducted a proof-of-concept study with six subjects (two female, four male). The accuracy of the system mainly depends on the lighting conditions and on the third party products used, namely the eyetracker and the IMU. Therefore, our primary goal was a qualitative evaluation of the visio-haptic interface. We also measured how well the system performed and which control mode was preferred in an obstacle avoidance task compared to a task without obstacles.

A. Procedure, tasks and measures

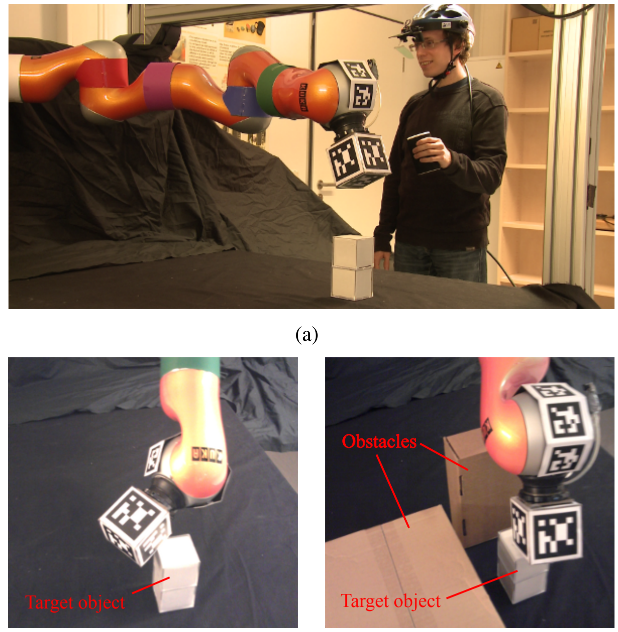
After a written and oral introduction, the participant put on the eyetracker helmet. The system was calibrated and initialized. Then, the participant had time to familiarize himself with the robotic arm and the control interface. Afterwards, the robot was moved to a defined starting position.

The subject's task was to navigate the cube attached to the robot's end effector to a target object which was positioned on a table. The task was finished when the end effector cube touched the target object. The subject performed two runs. In the first run, there were no obstacles. In the second run, two boxes had to be avoided. The obstacle avoidance task could not be accomplished without using the direct joint control mode. Figure 5(a) shows a participant operating the robot. The setups for both experimental conditions are depicted in Figures 5(b) and 5(c).

We measured the time the participant needed to complete the tasks and how often specific segments of the robotic arm were selected for moving. Furthermore, the subject was asked to fill in a questionaire. The questions were on prior experience with robots, usability of the visio-haptic interface and how hard or easy the subject perceived the tasks. Also, the participant was interviewed about his subjective experience with the system.

B. Objective Results

All subjects successfully navigated the end effector cube to the target object in both experimental conditions. However,



(a)

(b) (c)

Fig. 5. Experimental setup of the user study. (a) Participant operating the robot with the visio-haptic interface. (b) Setup without obstacles. (c) Setup with obstacles.

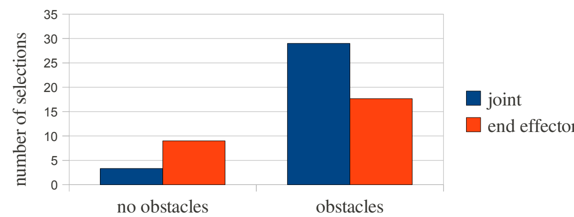


Fig. 6. Gaze-selections during the user study tasks. The amount of joint and end effector selections is plotted against the task condition.

one subject touched one of the boxes in the obstacle avoidance condition. The average time needed for the completion of the no-obstacle task was $72.5s$ ($SD = 93.8s$) and $208.1s$ ($SD = 123.8s$) for the obstacle avoidance task. This difference in duration between the conditions is not significant ($t(5) = 2.08; p = 0.09$).

In the no-obstacle condition, the subjects made, on average, 3.3 joint selections and 9 end effector selections by gaze. During the obstacle avoidance task, in contrast, single joints were selected more often (29 times) than the end effector (17.7 times). The results are visualized in Figure 6. Expressed as percentages, direct joint control was chosen significantly more often ($t(5) = 3.58; p < 0.05$) in the obstacle avoidance condition (69.8 percent, $SD = 17.8$ percent) than in the no-obstacle condition (30.3 percent, $SD = 32.4$ percent).

C. Subjective Results

The degree of familiarity with robots varied largely among the subjects. However, no influence on the performance could be noticed. In general, the subjects rated both gaze-selection and robot control with the *iObject* as "intuitive". Only half of the subjects experienced the audio feedback as "helpful" or "very helpful". One participant rated the obstacle avoidance task as "very difficult" and the no-obstacle task as "neutral". All other subjects found the navigation task without obstacles "very easy" and rated the obstacle avoidance task as "easy" or "neutral".

Some of the subjects said they found the end effector control inaccurate or did not like the two second limit. Others would have wished to rotate the *iObject* with their fingers or with both hands because they found it tedious to keep the object pressed during the whole rotation process. Also, it happened in a few cases that the gaze-selection failed. Haptic or visual feedback was suggested as an alternative to audio. One participant remarked that the combination of color and BCH code markers confused him a little. Another subject said that sometimes he was not sure about the direction of joint rotation. Nevertheless, most subjects expressed that they liked to operate the robot with the visio-haptic interface.

D. Discussion

The results of the proof-of-concept study show that the visio-haptic interface enables untrained subjects to control a robotic arm and to accomplish simple navigation tasks. Both operating modes were used by the participants. The results confirm that direct joint control is especially helpful for obstacle avoidance.

Increasing the speed of robot joint rotations could mitigate the tediousness issue. External references would probably improve the position estimation accuracy in end effector control mode but might decrease the mobility of the system. Gaze-selection failures dramatically reduce the usability. Therefore, it is worthwhile improving both the eyetracking accuracy and the computer vision approach. Acoustic feedback might not be the best way to convey the system state to the user. An alternative solution would be to equip the *iObject* with tactile stimulators. Generally, the capabilities of tangible interfaces like the *iObject* could be used for even more complex remote manipulation tasks like controlling a gripper mounted at the end effector.

VII. Conclusion

The objective of our work was to develop a visio-haptic human-machine interface to operate a complex robotic arm in a natural way. This goal was achieved by combining input from a mobile eyetracker and a tangible object called *iObject*. The presented system exploits human eye-hand coordination by directly mapping *iObject* movements onto movements of gaze-selected robot segments. Our proof-of-concept study showed that naive users perceived the interface as intuitive and easy to use.

Although this system is only a first prototype, there is potential for future applications in industrial or disaster robotics. In our current setup, the user and the robot are located in the same room. However, gaze-based segment selection is also possible from a screen image showing a camera stream, so that our basic approach is equally suitable for teleoperation scenarios. Future versions of the interface should incorporate even more haptic features replacing auditory by tactile feedback.

Acknowledgment: This work was partially funded by the German Research Council (DFG), grant EXC 277.

References

[1] M. Schröder, C. Elbrechter, J. Maycock, R. Haschke, M. Botsch, and H. Ritter, "Real-time hand tracking with a color glove for the actuation of anthropomorphic robot hands," in *IEEE-RAS International Conference on Humanoid Robots*, 2012.

[2] D. Kim, J. Kim, K. Lee, C. Park, J. Song, and D. Kang, "Excavator tele-operation system using a human arm," *Automation in Construction*, vol. 18, no. 2, pp. 173–182, 2009.

[3] M. R. Pedersen, C. Høilund, and V. Krüger, "Using human gestures and generic skills to instruct a mobile robot arm in a feeder filling scenario," in *International Conference on Mechatronics and Automation*, 2012, pp. 243–248.

[4] A. Finke, N. Hachmeister, H. Riechmann, and H. Ritter, "Thought-controlled robots - systems, studies and future challenges," in *IEEE International Conference on Robotics and Automation*, 2013.

[5] R. S. Johansson, G. Westling, A. Bäckström, and J. R. Flanagan, "Eye-hand coordination in object manipulation," *The Journal of Neuroscience*, vol. 21, no. 17, pp. 6917–6932, 2001.

[6] M. C. Bowman, R. S. Johansson, and J. R. Flanagan, "Eye-hand coordination in a sequential target contact task," *Experimental Brain Research*, vol. 195, no. 2, pp. 273–283, 2009.

[7] H. Koesling, M. Zoellner, L. Sichelschmidt, and H. Ritter, "With a flick of the eye: Assessing gaze-controlled human-computer interaction," in *Human Centered Robot Systems*, ser. Cognitive Systems Monographs, H. Ritter, G. Sagerer, R. Dillmann, and M. Buss, Eds. Springer, 2009, vol. 6, pp. 83–92.

[8] A. Uribe, B. Perez-Gutierrez, and S. Alves, "Gesture-based teleoperation using a holonomic robot," in *International Conference on Control, Automation and Systems*, 2012, pp. 208–213.

[9] P. Neto, J. N. Pires, and A. P. Moreira, "Accelerometer-based control of an industrial robotic arm," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2009, pp. 1192–1197.

[10] C. Smith and H. I. Christensen, "Wiimote robot control using human motion models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5509–5515.

[11] D. Zhu, T. Gedeon, and K. Taylor, "Exploring camera viewpoint control models for a multi-tasking setting in teleoperation," in *SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 53–62.

[12] R. Atienza and A. Zelinsky, "Intuitive human-robot interaction through active 3d gaze tracking," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, P. Dario and R. Chatila, Eds. Springer, 2005, vol. 15, pp. 172–181.

[13] H. O. Latif, N. Sherkat, and A. Lotfi, "Teleoperation through eye gaze (telegaze): A multimodal approach," in *IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 711–716.

[14] A. Úbeda, E. Iáñez, J. M. Azorín, J. M. Sabater, N. M. García, and C. Pérez, "Improving human-robot interaction by a multimodal interface," in *IEEE International Conference on Systems Man and Cybernetics*, 2010, pp. 3580–3585.

[15] R. Kõiva, R. Haschke, and H. Ritter, "Development of an intelligent object for grasp and manipulation research," in *International Conference on Advanced Robotics*, 2011, pp. 204–210.

[16] J. Wienke and S. Wrede, "A middleware for collaborative research in experimental robotics," in *IEEE/SICE International Symposium on System Integration*, 2011, pp. 1183–1190.

[17] M. Schöpfer, F. Schmidt, M. Pardowitz, and H. Ritter, "Open source real-time control software for the kuka light weight robot," in *World Congress on Intelligent Control and Automation*, 2010, pp. 444–449.

[18] K. J. Walchko and P. A. C. Mason, "Inertial navigation," in *Florida Conference on Recent Advances in Robotics*, 2002.

[19] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for gps-denied environments," in *IEEE International Workshop on Safety, Security, and Rescue Robotics*, 2007, pp. 1–6.