2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Evaluation of a Method for Intuitive Telemanipulation based on View-dependent Mapping and Inhibition of Movements

Simon Notheis, Björn Hein and Heinz Wörn

Abstract— **In this paper we present a novel approach for intuitive telemanipulation in Cartesian space and discuss the results of a user study evaluating different aspects of our approach. The proposed method inhibits certain degrees of freedom based on the current viewpoint. Together with automatic mapping of the input device to corresponding motion axes, our approach provides a very intuitive method for controlling the telemanipulation system while reducing the mental workload of the operator and therefore the amount of erroneous commands. Similar principles apply for controlling the viewpoint of real or virtual cameras to facilitate manipulation or navigation tasks.**

I. INTRODUCTION AND RELATED WORK

Telemanipulation, or teleoperation in general, covers a growing number of research fields and applications, especially when dealing with dangerous or very remote places. Familiar examples are robots exploring distant planets, search and rescue operations in disaster areas or defusing landmines from a safe distance. Teleoperated maintenance of off-shore drilling platforms may reduce the cost and time needed to transport personnel. Telesurgery provides quick access to medical expertise over large distances.

Two major challenges of all teleoperation systems are the restricted bandwidth and the delay in the communication channel. A common approach to deal with latency issues is to decouple the operator and remote robot by adding either autonomous skills or a model of the system that can be used for prediction and direct feedback to the user.

Therefore, one possibility for classifying teleoperation systems is the level of autonomy. Between the two extrema (fully autonomous and direct control) the most common categories are supervisory control and model-mediated control. The latter is often implemented by directly generating visual or haptic feedback for the operator based on a system model and the operator's input instead of waiting for the delayed actual feedback from the remote system (commonly known as predictive display or quickening) [1].

In many systems the teleoperation process is divided into different phases that reflect different levels of autonomy, e. g. a free motion phase with direct control, constrained motion control where only specific parameters or degrees of freedom are directly operated by the user while the system controls the others, or certain conditions that result in the rest of the task being executed autonomously [2][3].

Examples for these kinds of constrained movements are virtual walls or virtual fixtures that can guide an operator or block certain areas of the robot's workspace [4][5]. In contrary, our work will not limit the maneuverability of the manipulator but only the degrees of freedom that have to be controlled by the operator simultaneously at a given moment.

Another important area of investigation is the user interface design. The time and accuracy of completing a teleoperation task depend on how information and changes in the environment are presented to the user [6], as well as the user's situation awareness [7][8][9][10].

A common problem regarding the user interface is known as the "matrix of confusion" [11][12]. It refers to the mapping of input directions (e. g. given by the user via joystick and expected to be in his/her local coordinate system) and the resulting direction of the manipulator's movement (executed wrt. the manipulator's coordinate system). A fixed mapping together with the option to change between different cameras or viewpoints may result in rather unexpected behavior or can at least be considered error-prone since the user is responsible of taking account of the implicit coordinate transformations [13].

Many works try to evade the problem by using augmented reality (AR) to visualize color-coded coordinate systems or label corresponding axes of the input device [14][15]. Although these methods allow the user to look up the fixed mapping between input and motion axes, the mapping may not be intuitive wrt. the user's expectations and input controls may still involve a high degree of freedom for allowing Cartesian motion. Our approach will adjust the input device mapping according to the currently selected reference coordinate system and viewpoint from which the user observes the robot while also reducing the degrees of freedom necessary to control the system, as already mentioned above.

The telemanipulation system presented in this paper provides an AR user interface assisting the operator during the task by increasing the situation awareness. In addition to live sensor data and virtual models, our system focuses on intuitive methods for facilitating direct manipulator control in Cartesian space. Our novel approach considers and extends concepts like the matrix of confusion and allows the user to easily adjust the viewpoint to reduce the amount of erroneous commands.

The remainder of this paper is organized as follows: Section II summarizes previous and current system setups and will introduce some basic concepts behind our work. Section III will motivate and explain the chosen method for controlling the manipulator. In Section IV we will present and discuss evaluation results of a user study. The paper ends with a short conclusion of the presented methods.

S. Notheis, B. Hein and H. Wörn are with the Institute for Process Control and Robotics (IPR) at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, name.surname@kit.edu

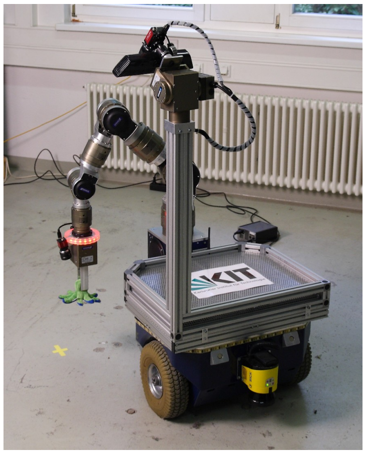

Fig. 1. Mobile robot platform “SR1” with 7-DOF arm, two-finger gripper and a Kinect camera on a pan/tilt sensor head.

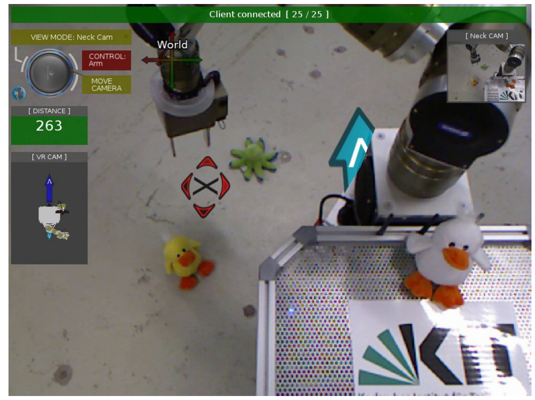

Fig. 2. AR cues (driving vs. viewing direction, TCP projected onto underlying objects/floor), meta information (obstacle distance, system status) and additional camera views rendered into the camera image for enhancing depth perception and situation awareness.

II. System Overview

A. Previous Work and Scenarios

In [16] we already presented a software framework for teleoperating (industrial) robots that included a CORBA-based communication and component model, as well as an augmented reality (AR) user interface. We also presented some intuitive methods for controlling pan/tilt and robot-guided cameras in the scene and autonomously executed high-level skills for recognizing and grasping objects.

In [17] we transferred the concepts and experiences from the industrial robot scenario to a mobile service robot platform (“SR1”, see Fig. 1) comprising a wheeled base, a 7-DOF arm and a pan/tilt sensor head equipped with a RGBD camera (Kinect). Compared to the industrial scenario, the major challenges include continuous grabbing of environmental data (point clouds), higher latency and reduced bandwidth due to WiFi and extended calibration and localization effort.

In addition to the semi-autonomous grasping skills that were adapted to the new hardware setup, we focused on investigating intuitive methods for directly controlling the manipulator in Cartesian space.

In [17] we introduced the concept of the “3rd person camera” for intuitive control of real and virtual cameras and derived a method for view-dependent inhibition and mapping of movements to facilitate Cartesian manipulator control. The basic ideas behind these concepts will be repeated in the following subsections before presenting evaluation results.

B. Design Guidelines

Basic features of our teleoperation system include:

- Collision avoidance by keeping an up-to-date 3D model of the robot and all known parts of the environment and continuously calculating obstacle distances to monitor movements of kinematics and objects.
- AR-based approach providing direct visual feedback for all commanded target positions at the remote user’s site (model-mediated predictive display).
- Augmentation of information such as obstacle distance, projected tool center point (TCP), driving and viewing directions, additional real/virtual camera views and other status information (see Fig. 2).

The main goals of the manipulator control strategy presented in this paper are:

- Reducing the user input complexity/DOF to reduce the mental workload and erroneous command inputs.
- Inhibition of movements that are dangerous or difficult to assess (e. g. due to the user’s current point of view).
- Intuitive mapping of allowed movement directions to the input device to preserve the user’s expectation regarding the resulting movement.

These goals and their implementation will be described in more detail in the following section.

III. Manipulator Control

The following subsections will explain our concept for intuitive manipulator control, starting with an easy-to-use method for controlling real and virtual cameras. Based on that, we will motivate the inhibition of certain degrees of freedom and describe its implementation. The last subsection focuses on the intuitive mapping of input and motion commands.

A. Camera and Viewpoint Control

Since our approach for inhibiting and mapping movements depends on the current viewing direction, we will first explain how the viewpoint can be controlled. In normal navigation mode, the user will most likely stick to the camera view and image stream provided by the sensors of the pan/tilt sensor head. Its two degrees of freedom can easily be controlled by any joystick-like input device or by directly mapping the user’s head movements when using a head-mounted display (HMD) with motion/inertial sensors.

The main disadvantages of systems using only video streams from real cameras are the limitations in movability of the camera resulting in occlusion of certain areas and the difficult perception of depth. Although the latter may

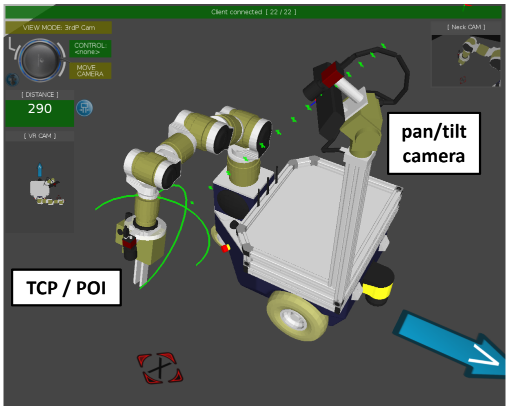

Fig. 3. While the field of view of the real pan/tilt camera is limited, the virtual 3rd person camera can be positioned freely in the scene (with only three degrees of freedom to control) by defining a polar coordinate system around a given point of interest (POI).

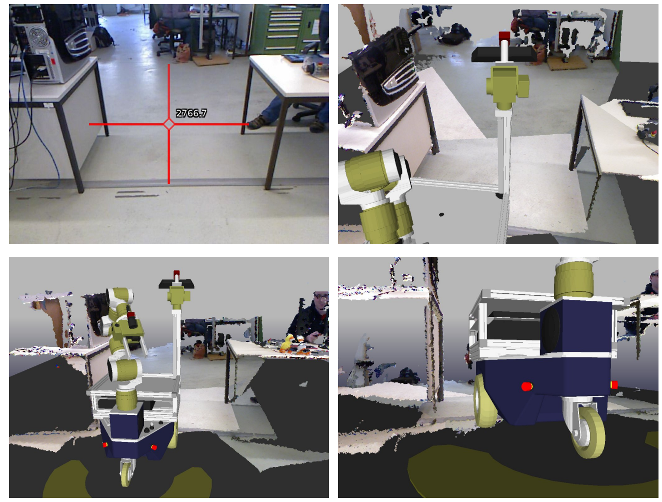

Fig. 4. Use of 3rd person camera when navigating through narrow spaces.

be improved by using stereoscopic cameras and displays, a free choice of a viewpoint may still be desirable in certain situations. Given the continuous 3D modeling of the environment, it is obvious that we can easily change the viewpoint from the real camera to any virtual camera position in the environment model.

Following our first guideline, to reduce the input complexity, the user won't be required to control all six degrees of freedom of the virtual camera. Instead, our so-called "3rd person camera" approach will reduce the input complexity to one degree of freedom for zooming and two angles defining the orientation of the camera that will always be focused on a given point of interest (POI) derived from the current task.

For manipulation scenarios the POI can be set to the gripper/TCP (see Fig. 3) or to the object that is to be grasped. The virtual camera can then easily be moved around the POI. In our previous industrial robot scenario we used the same technique to position a real camera carried by an industrial robot (KUKA KR-16) while observing a second robot executing semi-autonomous grasping tasks. Since in that scenario the distance (zoom) to the POI was fixed, the remaining two angular degrees of freedom could be controlled hands-free via head movements recognized by the HMD's inertial sensor, instead of using a 6-DOF input device to position the robot's end-effector manually in Cartesian space.

For navigation tasks or without a specific goal, the POI resides in front of the real pan/tilt camera mounted on the sensor head. This way, the operator can seamlessly "zoom out" of the actual camera view into the virtual camera view, observe the environment from behind the robot or place the virtual camera beside or above the robot to cope with the difficulties of navigating through narrow areas (see Fig. 4).

B. Inhibition

The inhibition of certain movements or Cartesian axes when controlling the manipulator can have various reasons and manifestations. When using a 6-DOF mouse, for example, it is difficult for a non-experienced user to perform exact movements along one translational axis without translational or rotational components in the other axes. Therefore it may be desirable to lock certain axes or to switch between two modes that allow only translational or only rotational command output respectively.

Another reason for inhibiting certain movements may be the definition of virtual boundaries either based on collision avoidance algorithms or task specific guidance. The inhibition of certain degrees of freedom may result in a complete blockage or a scale-down in speed.

We furthermore postulate that, given a certain view of the manipulator, it is difficult to assess movements along certain axes due to a lack of depth perception, occlusion, or due to less visual changes of the resulting movement in the projected image. In particular, translational movements parallel to the optical axis are much harder to assess than translational movements parallel to the image plane. With respect to rotations, the degree of visual change depends on the geometry and relative orientation of the manipulator, but it can be assumed that rotations about an axis parallel to the optical axis will generally result in the most visible change.

Regarding the inhibition of movements, we therefore decided to reduce the available degrees of freedom for controlling the position and orientation of the manipulator to only three of six possible ones: two translational axes (the ones with the smallest angle compared to the image plane) and one rotational axis (the one with the smallest angle wrt. the optical axis of the current view).

The reference coordinate system (CS) itself, of which the axes to block and to allow movements along are chosen, may hereby vary depending on the given task or user's choice. Useful reference coordinate systems include: world and/or mobile platform base CS, the manipulator's TCP CS and the camera CS. To perform movements along a currently blocked axis (e. g. translation away from the camera) the user is forced to change the viewpoint. In our opinion, the benefits

(i. e. lowering the mental workload and risk of undesired lateral movements by reducing the input complexity and having a better view for assessing the commanded movements) outweigh the restriction in movement and the need to re-position the camera more frequently, that more experienced users may complain about.

C. Mapping

To optimize the mapping between the allowed motion axes and the input device, we resort to compensating the "matrix of confusion" or, loosely speaking, the concept of "up is up and left is left". Therefore the mapping of axes changes according to the relative position and orientation between the viewpoint and the chosen reference coordinate system for the manipulator movements in such a way, that the direction of the resulting movement will meet the user's intuition about what is left, right, up and down wrt. the reference coordinate system's projection in the current view (see illustrations in the right column of Fig. 5).

The left column of Fig. 5 shows two examples of possible restricted movement configurations. In the first image, the user may issue movements towards/away from the table (left/right) or in the vertical direction (up/down). Rotation is only allowed about the third axes. Movements towards/away from the camera are blocked because they are more difficult to assess from this point of view. In the second image the user changed the viewpoint to a position above the robot. The adapted coordinate mapping now allows translational movements parallel to the table's surface. Again, translations along the optical axis (resulting in vertical manipulator movement wrt. the world coordinate system) are blocked. The current mapping is indicated by arrows rendered into the image at the end-effector's position. Fig. 6 shows snapshots of an actual grasping process using this approach.

IV. Evaluation and Discussion

A. Overview and Setup

The telemanipulation task given to the participants was to pick up an object from the floor and to drop it onto the loading area of the mobile robot platform shown in Fig. 1. Each participant had to complete the grasping task twice: In the first run, the manipulator, real and virtual camera were controlled via a 6-DOF mouse, in the second run, our reduced 3-DOF concept was applied and a gamepad was used as input device. It should be noted though, that the focus of the evaluation was the comparison between the classic 6-DOF and our 3-DOF approach and not the comparison of input devices, as will be discussed in more detail later in this section.

The robot system and the user's client application were communicating via a LAN connection, transmitting a live video stream and point cloud data from the Kinect mounted on the pan/tilt sensor head. The user can switch between the actual camera view where he/she can also control the pan/tilt unit of the sensor head, and the virtual camera that can be positioned rather freely in the scene.

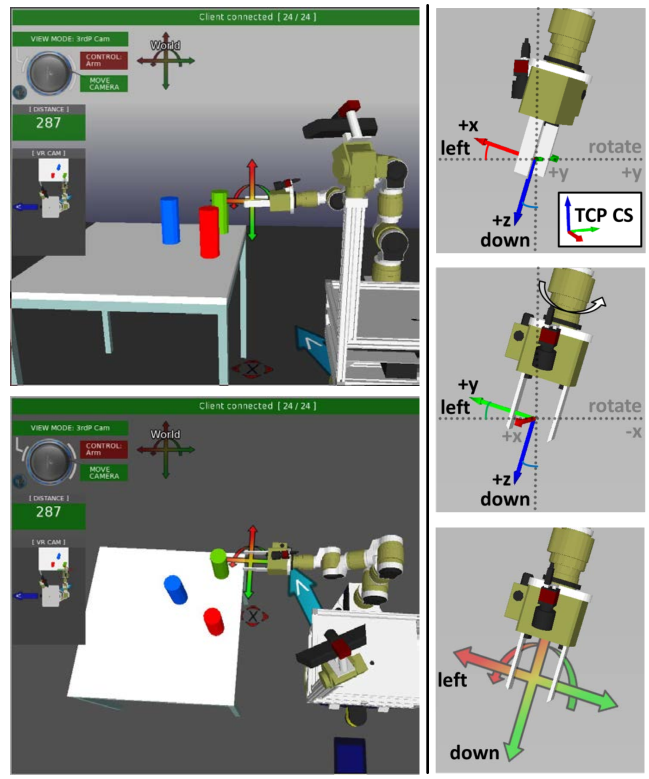

Fig. 5. Two examples for view-dependent selection and mapping of allowed movement axes. The system allows two translational and one rotational degree of freedom for controlling the manipulator. Translation is only possible along axes that form an acute angle with the image plane since the resulting movement can be assessed best from the current point of view.

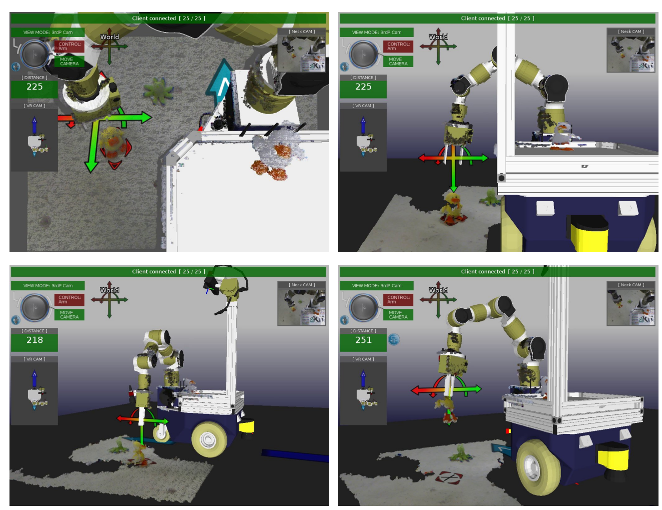

Fig. 6. Demonstration of view-dependent inhibition and mapping of movements during telemanipulation. The images are snapshots from a live video of the operator's GUI.

As already indicated in Fig. 2 and Fig. 6, different kinds of information are rendered into the user's GUI and camera view, including a downwards projection of the current TCP position and a second, semi-transparent manipulator visualizing the commanded target position (not shown in the figures) while the movement of the actual manipulator might be delayed due to restrictions in velocity or due to latency in communication. This predictive "shadow manipulator" also indicates if the desired target position is unreachable due to restrictions of the kinematic's workspace or due to collision at or on the path towards the target position. During manipulator control, the current reference coordinate system and mapping of the axes is also rendered into the user's view.

Before controlling the real robot, each participant got an introduction to the concepts and input device button mapping of about 30 minutes, including the opportunity to practice the control with a simulated robot.

B. Implementation Details

As mentioned before, our 3-DOF control concept not only includes the inhibition of certain axes based on the current viewpoint, it also compensates the matrix of confusion by providing an intuitive mapping between the chosen reference coordinate system for the manipulation and the user's expectation regarding the mapping of the input device to his/her current view of the manipulator.

To ensure a fair comparison, we therefore made some adaptations to the classic 6-DOF control approach. For controlling the virtual camera, we locked the roll angle, since in most cases it just doesn't make much sense to rotate the camera about the optical axis. For 6-DOF manipulator control, we partly compensated the matrix of confusion, but only regarding the global z-axis, i. e. when the virtual camera is rotated around the manipulator, the axes for moving it left/right and towards/away from the camera are adapted to meet the user's expectations while still allowing full 6-DOF motion control.

The GUI elements representing the currently active view and control mode, as well as the current minimal distance of the manipulator to other objects in the environment, are colored green/yellow/red according to the necessary level of the operator's attention (low/medium/high respectively). Nevertheless, due to online collision avoidance, it was not possible to crash into any known obstacles.

As an optional feature, automatic camera guidance could be used so that the pan/tilt sensor head always follows the TCP, which eliminates the need of switching between the real and virtual camera, but restricts the sensor data to the surrounding of the TCP instead of an area chosen by the operator.

C. Questionnaire Results

After completing the task in both 6-DOF and 3-DOF control mode, the 20 participants were asked to state their opinion in a short questionnaire. The distribution of their ratings is shown in Table I. Additional free text comments could be added as well, some of which will also be discussed here. Most participants were researchers in the area of computer science, mechatronics, medical, industrial or micro robotics, but with very different background in robot control.

All participants concurred that the option of switching to a virtual camera view, visual hints like the projected TCP position and the augmentation of the "shadow manipulator" showing the commanded target position along with its reachability, were extremly helpful, since they improved depth perception and situation awareness and prevented the common move-and-wait strategy seen in many teleoperation systems with latency. Whereas the color-coding of GUI elements according to the necessary level of attention was appreciated by the users but rarely paid attention to during the task. But this may differ in other scenarios where the user has to switch between various view and control modes more often than during the task given here.

TABLE I

RESULTS OF THE QUESTIONNAIRE: DISTRIBUTION AND ROUNDED MEAN μ OF THE RATINGS FOR ALL 20 PARTICIPANTS

	-2	-1	0	+1	+2	μ
Experience in						
– using 6-DOF mice	–	–	9	7	4	0.7
– using 6-DOF haptic devices	–	–	9	10	1	0.6
– using gamepads	–	–	5	10	5	1.0
– coordinate system transformations	0	0	3	5	12	1.5
(0 = none, +2 = very experienced)						
Usefulness of						
– AR cues (e. g. TCP projection)	0	0	0	1	19	2.0
– color-coded GUI elements	0	0	8	9	3	0.7
– Automatic camera guidance	0	0	2	12	6	1.2
(-2 = very confusing, +2 = very helpful)						
3-DOF control						
– Intuitiveness	0	0	1	9	10	1.4
– Confidence	0	0	0	7	13	1.7
3-DOF vs. 6-DOF control						
– Accessibility	0	1	4	11	4	0.9
– Intuitiveness	0	1	2	11	6	1.1
– Confidence	0	0	1	7	12	1.6
(-2 = low/worse, +2 = high/better)						

For evaluating our 3-DOF control concept, the participants were asked to assess the subjective intuitiveness of controlling the camera and the manipulator, as well as their confidence while commanding the robot, which includes their situation awareness. In general, our 3-DOF approach was deemed more intuitive and safer than the 6-DOF approach. Contrary to our expectations, we could not derive a clear correlation between previous experience in using 6-DOF input devices and the assessment of our approach in comparison to the 6-DOF control mode.

According to our expectations, many participants noted, that in our view-dependent 3-DOF approach, they were forced to change the viewpoint much more often in order to access currently blocked movement directions. On the other hand, they felt safer when commanding the robot in 3-DOF mode, since they knew, that certain involuntary rotations or translations were blocked. Also, full 6-DOF control seems to result in smoother and maybe faster movements, but the operator may constantly be making corrections, e. g. regarding involuntary rotations. Especially in scenarios where full collision avoidance can not be guaranteed, most operators will prefer safety over more degrees of freedom.

The comparison in accessibility relates more to the choice of the input device itself. Given a basic understanding of the working principle of a 6-DOF mouse, it may seem more accessible (independent of the actual ease of use), since classic gamepad controller elements like analog sticks cannot control more than two degrees of freedom at the same time. In our case, some participants criticized the mapping of both manipulator rotation and camera zoom (depending on the active control mode) to the analog shoulder buttons, whereas the other two degrees of freedom for manipualtor and camera control were clearly separated by using the left and right analog stick respectively. Accessibility is also influenced by previous experiences with the specific devices.

D. Log File Evaluation

During task execution, log files were created for each participant containing various information regarding the current state and user inputs.

Comparing the mean task completion time for 3-DOF mode (μ = 211 s, σ = 84 s) and 6-DOF mode (μ = 285 s, σ = 90 s), one can observe that with our approach, task completion time could be decreased by 26 %. Looking at the mean time improvement for the individual participants, our 3-DOF approach is even 33 % faster. Although no clear correlation between the difference in task completion time and previous experience with the input devices could be derived, this may change if the users become more familiar with the system and 6-DOF control.

Although many users noted the need to change the viewpoint more often in the 3-DOF run, the logs show that the actual time spent for manipulator control was almost the same (54 % vs. 52 %), most probably because in 3-DOF mode the virtual camera could be positioned more precisely and therefore faster. Regarding the overall mean values, the virtual camera covered twice the distance and did almost three times the rotations in 3-DOF mode as compared to 6-DOF mode.

As expected, the rotational part of manipulator commands increased significantly (by a factor of 5) in 6-DOF mode, due to involuntary rotations about the other non-inhibited axes.

Since one of the basic ideas behind our 3-DOF concept was to keep manipulator movements rather perpendicular to the virtual camera's optical axis for better assessment, we also analyzed the angle between the current camera orientation and the commanded target direction during manipulator control. The mean deviation from the preferred value of 90° yielded 19° in 3-DOF mode and 34° for 6-DOF control. One might argue that in 6-DOF mode, movements towards the camera can be assessed much better than movements away from the camera, but considering the angular deviations separately yields the approximately same result. That means that the users did not deliberately prefer movements towards the camera over movements away from or perpendicular to the camera.

V. Conclusion

We introduced a novel concept for view-dependent inhibition and mapping of motion axes for Cartesian telemanipulation applications. Our user study with 20 participants has shown, that the approach is deemed more intuitive, provides better situation awareness and may result in faster task completion than classic 6-DOF control concepts.

References

[1] A. Rodríguez, E. Nuño, L. Palomo, and L. Basañez, "Nonlinear control and geometric constraint enforcement for teleoperated task execution," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5251–5257.

[2] G. Lian, J. Cui, Z. Qingjie, J. Zhipu, and Z. Sun, "Telemanipulation via internet based on human-robot cooperation," in *Proc. of 2001 IEEE International Conferences on Info-tech and Info-net ICII*, vol. 4, 2001, pp. 256–262.

[3] N. Ouramdane, F. Davesne, S. Otmane, and M. Mallem, "3d interaction technique to enhance telemanipulation tasks using virtual environment," in *Proc. of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006, pp. 5201–5207.

[4] E. Damdinsuren, K. Kosuge, Z. Wang, and Y. Hirata, "Virtual elastic wall based motion control for teleoperated demining system," in *Proc. of 2005 IEEE International Conference Mechatronics and Automation (ICMA)*, 2005, pp. 1666–1671.

[5] S. Payandeh and Z. Stanisic, "On application of virtual fixtures as an aid for telemanipulation and training," in *Proc. of 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems HAPTICS*, 2002, pp. 18–23.

[6] P. Mitra, D. Gentry, and G. Niemeyer, "User perception and preference in model mediated telemanipulation," in *Proc. of 2007 IEEE EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC)*, 2007, pp. 268–273.

[7] M. R. Endsley, "Design and evaluation for situation awareness enhancement," in *Proc. of 32nd Human Factors and Ergonomics Society Annual Meeting*, 1988, pp. 97–101.

[8] H. A. Yanco, J. L. Drury, and J. Scholtz, "Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition," in *Humam.-Computer Interaction*, vol. 19 (1), 2004, pp. 117–149.

[9] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, "Ecological interfaces for improving mobile robot teleoperation," in *IEEE Transactions on Robotics*, vol. 23 (5), 2007, pp. 927–941.

[10] D. G. Caldwell, K. Reddy, O. Kocak, and A. Wardle, "Sensory requirements and performance assessment of tele-presence controlled robots," in *Proc. of 1996 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 1996, pp. 1375–1380.

[11] E. C. Morley and J. R. Wilson, "The matrix of confusion - a classification of robot movement," in *Journal of Engineering Manufacture*, vol. 210 (3), 1996, pp. 251–260.

[12] J. A. Macedo, D. B. Kaber, M. R. Endsley, P. Powanusorn, and S. Myung, "The effect of automated compensation for incongruent axes on teleoperator performance," in *Human Factor*, vol. 40 (4), 1998, pp. 554–568.

[13] A. Nawab, K. Chintamani, D. Ellis, G. Auner, and A. Pandya, "Joystick mapped augmented reality cues for End-Effector controlled teleoperated robots," in *Proc. of 2007 IEEE Virtual Reality Conference VR*, 2007, pp. 263–266.

[14] S. Hashimoto, A. Ishida, and M. Inami, "Touchme: An augmented reality based remote robot manipulation," in *Proc. of International Conference on Artificial Reality and Telexistence (ICAT)*, 2011.

[15] K. Chintamani, A. Cao, R. D. Ellis, and A. K. Pandya, "Improved telemanipulator navigation during display-control misalignments using augmented reality cues," in *IEEE Transactions on Systems, Man and Cybernetics*, vol. 40 (1), 2010, pp. 29–39.

[16] S. Notheis, G. Milighetti, B. Hein, H. Wörn, and J. Beyerer, "Skill-based telemanipulation by means of intelligent robots," in *Proc. of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 5258–5263.

[17] S. Notheis, B. Hein, and H. Wörn, "View-dependent mapping and inhibition of movements for intuitive telemanipulation," in *Proc. of 1st Workshop on Industrial Mobile Assistance Robots (IMAR)*, 2013.