UBIQUITOUS USER INTERFACE DESIGN FOR SPACE ROBOTIC OPERATION

Peter Birkenkampf, Daniel Leidner, and Neal Y. Lii

German Aerospace Center (DLR), 82234 Weßling, Germany

ABSTRACT

Autonomous robotic systems are approaching the maturity level to be able to support astronauts in space and on planetary surfaces. Commanding these robots with existing input devices and command modalities cannot cope with their increased capabilities and limits the usability for the astronaut. This paper presents alternative user interface (UI) concepts for the use with ubiquitous devices such as smartwatches and tablet computers. In particular, it is proposed to command an autonomous space robot assistant on a low to medium level of autonomy using a smartwatch. When commanding the robot on a high level of autonomy, a tablet computer can be utilized with object-centered task-level commands. The respective interaction concepts designed to make best use of the input devices are presented in detail. A comparison of the proposed interfaces and the presentation of the use of the tablet computer user interface in the METERON SUPVIS Justin ISS experiment concludes the paper.

Key words: Human Robot Interaction; Intuitive GUI Design; Knowledge-Driven GUI; Supervised Autonomy; Space Telerobotics; METERON SUPVIS Justin.

1. INTRODUCTION

Advanced robotic systems are great assets to the endeavors in space, such as on-orbit servicing and planetary exploration. The main tasks of these robots include exploration, maintenance of deployed structures, and setup of infrastructure for future missions. Autonomous features will enable these robots to accompany the astronaut as a co-worker rather than as a tool extension. This allows the astronaut to focus on other more immediate problems and work, with supervising the robot relegated to a side task. The novel way of collaboration between astronauts and robots raises the need for a new user interface to command the extended robot capabilities. Traditional input devices such as personal computers and notebooks are not practical for field operation with a robotic co-worker. In this paper, we propose to use common portable hardware devices, like tablets and smartwatches, for interfacing with the robotic coworker as visualized in Figure 1. The proposed interaction concepts allow the user to command the robot on different levels of autonomy. As an example, a low-level control modality would allow the user to set the power state of the mobile base. Medium-level command modalities include direct control of the robots gaze and the base position. High-level command modalities let the user command task-related actions to the robot, such as manipulating an object. The focus of this paper lies in the commanding of high- and medium-level autonomy features of the robot which qualify the robot to act as a coworker for the astronaut. We propose the deployment of portable, ubiquitous devices for interfacing with the robot that can be used individually or complementary. As the robots executes the commands autonomously, the resulting interfaces can be used from remote locations with delayed and unreliable communication links.

Figure 1. The humanoid robot Rollin' Justin from DLR maintaining a solar panel in a simulated Martian environment. It is commanded via a tablet computer complemeted by a smartwatch user interface.

The remainder of this work is structured as follows: In Section 2 We put this paper into the context of recent related work regarding ubiquitous UIs. Based on that, we summarize the command modalities we require from a robot to be used as a coworker in Section 3. We then present concepts for ubiquitous UIs for smartwatches in Section 4 and tablet computers in Section 5. We compare the interface concepts and show the benefits of complementary use of both UIs in Section 6. We conclude the paper in Section 7 with an outlook on the space validation of the proposed tablet computer UI concept in the METERON SUPVIS Justin ISS experiment.

2. RELATED WORK

The usage of common portable hardware devices for commanding robots is a need arising from the deployment of autonomous robots. Particular small interfaces can be realized using recently emerging smartwatches. Their main advantage is the convenient wrist-mount that makes them easy to carry, and always available for the user. A main problem when using smartwatches for commanding robots is their small touchscreen size allowing only for limited information display and interaction modes. One possible solution to this problem is the use of gestures to command robots rather than on-screen commands. Gesture-commanded robots have been successfully used to perform high dexterity manipulation tasks [LCR+12]. Recent investigations have also begun using the built-in sensors of the smartwatch to command mobile robots with gestures [VSR+17]. A different approach uses the smartwatch to visualize information and send touchscreen-selected commands to a robot [MS14]. This approach tries to solve the size-limitation of the screen by only showing one control at a time. The user has to use sliding gestures to select the desired interface from a list.

Due to the size limitation of smartwatches, it is more common to use larger devices like smartphones for commanding robots. Simple smartphone interfaces allow the user to command the robot using virtual buttons and joysticks or using the built-in sensors of the mobile device [LCK11][KHMK13][CBWC15]. These interfaces can be made more intuitive to use by displaying the live video of the robot and augmenting it with information about the robot and the environment [AHY12]. [AK16] provides landmarks in the live video to command the robot to travel towards the augmented regions autonomously.

Recent investigations in the use of portable hardware devices for commanding robots focuses on the use of tablet computers, which offer significantly larger screen sizes and higher resolution than smartphones. [MBGH14] implement a visual programming concept that allows non-expert users to program an industrial robot using a tablet computer. The main features of the proposed system are a simulated view of the robot for the visualization of teached paths and target poses, and an augmentation of the video of the robots with task-specific information during execution. A more direct approach of commanding robots is proposed by [MSB12]. The tablet computer is used to command a service robot on different levels of autonomy using virtual buttons and joysticks for directly moving the robot actuators on the lower levels and task-related commands on the higher levels. Augmented reality is used to highlight the estimated position of the objects that the robot can manipulate in the video of the robot. [FMK16] propose a UI for commanding pick-and-place tasks that uses no virtual buttons or joysticks at all. The object of interest is selected in the video of the camera of the tablets and drag-dropped to the target location. Using the camera of the tablet as a reference for commanding the robot is an intuitive approach as the user does not has to command the robot to look at the object of interest before commanding the manipulation. However, this approach is difficult to implement for space telerobotics applications as it is required to be in place with the robot.

In [BLB14] we proposed an interface for commanding not only pick-and-place but a wider array of tasks to the robot. The key idea is to utilize an object-centric approach, where each object affords possible interaction methods to the user. As such, commanding the robot comes down to simply selecting the object of interest in the video of the robots and choosing the desired action from a list of current possibilities. This direct and intuitive command interface of the robot on a high level of autonomy makes the proposed approach suitable for the use with a robotic coworker for space operations [LBLB14].

In this work, our previous approach is adapted for the application in an actual space robot experiment within the METERON project suite.

3. COMMAND MODALITIES FOR A ROBOTIC COWORKER

The robot interfacing with the proposed ubiquitous UIs aims to be used to take over repetitive and/or dangerous work or to be deployed at locations which the user cannot reach in person. To work as a proper functioning coworker for the user, a sufficient level of autonomous capability is required. As the robot should also be a useful partner when autonomy fails, commanding of low-level autonomy features is necessary as well. Therefore we propose a system with command modalities on a low, medium and high level of autonomy.

Low-level autonomy features hand over the complete control of the robotic hardware to the user such as changing the power state of the motors or commanding target velocities of the mobile base. These features may be helpful to resolve situations in which the autonomy of the robot failed. Most likely the user has to be familiar and in the surrounding of the robot to efficiently use these features. A drawback is, that these low-level features leave the user with the complete responsibility for the robot and its environment causing high stress levels when commanding the robot.

Medium-level autonomy features help the user by taking over safety considerations for the robot and its environment. On this level, the robot e.g. does not allow velocity commands by the operator if they could lead to a collision with a object of the environment of the robot.

High-level autonomy features allow the user to command task-like actions to the robot which are autonomously executed. In our previous work, we have shown that arranging the knowledge for these actions within an object-centered context is an efficient approach to solving manipulation tasks [LBH12]. Following this concept, we organize objects in a hierarchical structure according to their functionality.

The resulting knowledge system consists of three parts: an *object storage*, a *world representation*, and a *hybrid reasoning framework*. We collect the prior knowledge about all objects the robot should interact with in the *object storage* where we arrange the objects in the object-

oriented paradigm. This means that the objects of the same class share the same process models to interact with the environment. Specific properties such as size and shape can vary among the specialized classes. Specific instances of these object classes represent the current state of the environment and are composed in the *world representation*. The current state of the object instances is available through their class-specific symbolic and geometric properties. The *hybrid reasoning framework* uses these properties and the distinct manipulation instructions stored in class-specific *action templates* to evaluate the feasibility of actions symbolically. The resulting symbolic transition is afterwards geometrically evaluated to ground the symbolic actions into robot-specific skills by the use of modular geometric simulations such as navigation, motion planning or dynamics simulations.
We successfully evaluated this approach to solve everyday manipulation tasks [LBH12], mobile manipulation tasks [LB13], and force-sensitive whole-body manipulation [LDS+14].
In [BLB14] we present a method to generate a list of all current action possibilities from the current *world representation*. This list is used to generate object-specific robot commands that are available for the user.
We accompany this concept by a *mission control* component that opts-out actions that are not required to complete the current mission [LBL17]. By this, the number of possible robot commands is reduced to a manageable amount and the user can be guided in its decisions with respect to the mission objectives.
In this paper, we focus on using the robotic coworker as as mobile service robot. Possible tasks include observation, data collection, navigation, and manipulation.

4. SMARTWATCH UI CONCEPT

On different occasions such as extra-vehicular activities or field missions, it may be too cumbersome or impossible to carry a handheld input device, such as a tablet computer, for commanding the robot. Therefore we propose the use of a smartwatch as an example for a touchscreen device with minimal screen size and good portability. A watch can be worn by the user at little to no additional load carrying, while granting immediate access to the robot if needed.
As the input methods and display size of a smartwatch are rather limited, classic visualization and interaction concepts cannot be applied. A common approach is to use (nested) lists as navigational elements. The drawback of this method is, that the user can only see a few items at the same time on the small screen so the user might have to scroll through the list first before selecting an item. As the capabilities of the robotic coworker might change depending on the current status of the robot and its environment, it is important for the user to be able to see all current options to determine a reasonable procedure to fulfill the mission.

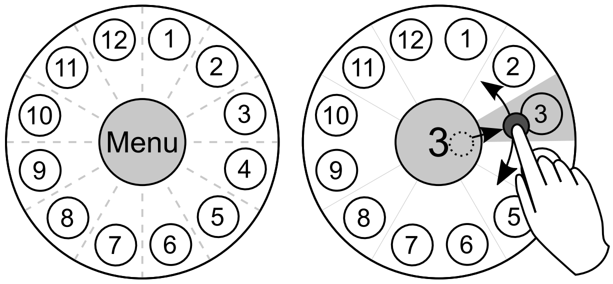


Figure 2. Conceptual circle menu for the use in the smartwatch UI. The screen is divided in equally-spaced segments with an interaction-starting center area (left). An item is selected by swiping from the center into the circle segment of the desired item (right).

4.1. Circle Menu

For the smartwatch concept we propose a circle menu to navigate in the smartwatch UI as shown in Figure 2. The screen is divided in circle segments, each segment representing a list item. The selection process begins by touching the center region of the screen and swiping outside into the circle segment containing the desired item. Releasing the screen inside a circle segment selects the related item while releasing in the center region selects nothing. With this input method, only the screen center is initially used for the menu interactions while other touch-inputs such as swipes from the border of the screen are still available (e.g. to switch views). However, after the initial touch was made, the whole screen is used for the item specification, making best use of the limited space of the smartwatch. The selection process is supported by tactile feedback by vibrating the smartwatch when a new item is selected.
The center region of the screen is allocated to show the title of the current menu or an enlarged version of the current icon of the item while selecting. This is of particular interest, as the currently selected item is occluded by the input finger, while the center region remains visible.
The usability of this menu has been tested for up to 8 item using text labels in the circle segments and 12 items using icons. Using more items in a menu makes it hard to recognize the icons on the small screen. A fix is to add a new layer to the circle menu providing the next set of items, which can be accessed by clicking the center region.

4.2. Interaction Concept

The proposed smartwatch UI for commanding service robots is tailored to the small screen size. This property of smartwatches must be carefully considered to be able to effectively display information about the status of the robot to the user. In particularly, displaying video is not considered helpful, as details of the environment of the robot are not visible on the small screen. Even though the video could be used to gain an rough overview of the environment, we have decided not to implement video

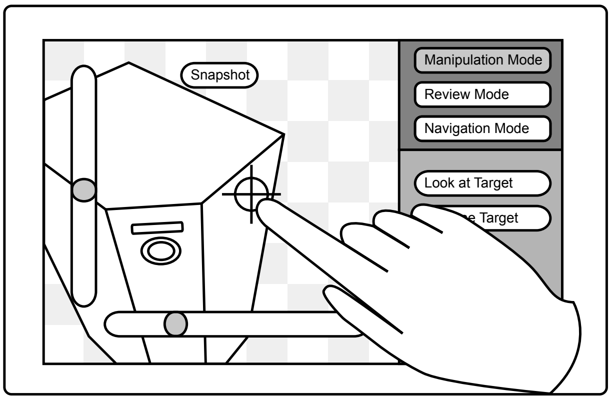


Figure 3. Conceptual sketch of the tablet computer UI. The screen is divided in the content (left), mode selection (top right), and command area (bottom right). The content area shows the current video of the camera of the robot. A gaze target can be specified using sliders or directly clicking in the camera image. Context-specific commands in the command area allow to move the robot accordingly, to look at the selected target.

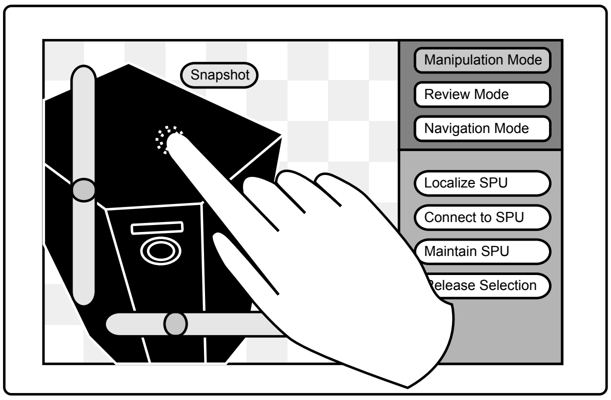


Figure 4. Concept of object manipulation commanding. The video stream of the robot is augmented with CAD models of the world state objects. By selecting an object in the video, the user specifies an interaction intent. The context-specific commands related to the selected object are then displayed in the commands area.

functionality on the smartwatch. There are two main reasons for this decision: First, it is not possible to show the video and command options simultaneously given the limited display area. The required constant changing of different UI views would render it too difficult to operate. Second, commanding the robot without noticing the details in the video could lead to errors, which could cause the user to select dangerous or wrong commands.
Without showing the video to the user, the applicability domains of our user interface are reduced to situations where the user is in close proximity to the robot and does the supervision on its own or the robot provides reliable autonomy to identify and avoid dangerous or wrong commands so no supervision is needed.
Especially when commanding a robot on a low level of autonomy, the user needs to assess the internal state of the robotic system. Therefore we provide telemetry information of the robot, such as power status of the actuators or the current high-level-controller, in a separate GUI view which can always be dragged over the current view by swiping down from the top border of the screen. Providing only the telemetry information is sufficient for the proposed interface because commanding a robot on a low level of autonomy requires a trained user who is familiar with the system.

The proposed UI concept consists of layered instances of the described Circle Menu. The first layer serves to select the level of autonomy of the robot commands. Low-level command modalities are exemplified by inputs allowing the astronaut to set the power state of the mobile base or changing the high-level controller. Medium-level command modalities include direct control of the gaze of the robot and the base position while utilizing collision avoidance functionality of the robot. The command modalities of the low and medium level are directly commanded using the corresponding Circle Menu for on/off options, layered Circle Menus for item selection options, virtual sliders for value selection options and virtual joysticks for movement options. Due to the direct commanding of the robot, the low- and medium-level modalities can only be used in a setup with a reliable communication link with minimal delay.
High-level command modalities allow the user to command task-related actions to the robot, such as traveling to a target or manipulating an object. Due to our object-specific approach for commanding robots, as described earlier, we propose a two-steps process to select a command. First, an object of the current world representation of the robot is selected in a Circle Menu. Then, the desired action belonging to the chosen action is selected in the next layer of the Circle Menu. With this design, we reduce the amount of information to show to the user to a minimum amount while providing full access to the autonomous capabilities of the robot. As the robot executes the commanded task autonomously, high-level commands are suited for setups with an unreliable communication link with high delay.

The proposed smartwatch UI allows the user to command a service robot on different levels of autonomy. The main advantage of the UI is its good portability caused by the small form factor and the wrist-mounted setup of the smartwatch. The main problem is the small screen size, which limits the visualization and input methods. This results in a UI which requires the user to be familiar with the system to be able to operate it efficiently.

5. TABLET UI CONCEPT

A tablet computer is utilized in this study as an example for an UI on a still portable device with a large screen. The screen allows us to show information about the status of the robot to the user while simultaneously providing a command interface. In this paper, we present an intuitive

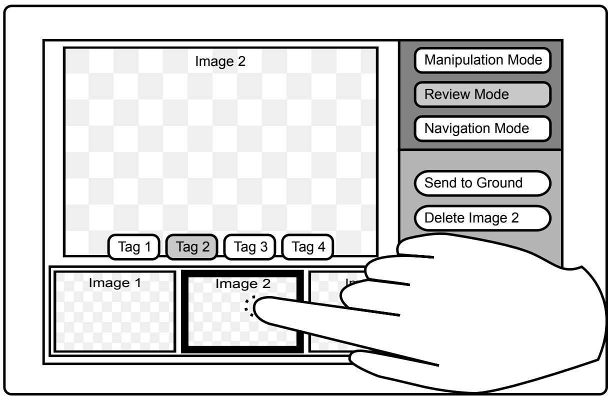


Figure 5. Interface concept for reviewing snapshots. A snapshot is selected in the image gallery slider at the bottom of the content area. Context-specific tags can be selected for each snapshot.

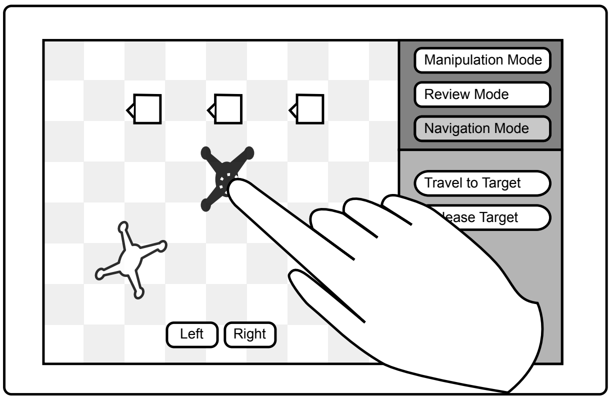


Figure 6. Interface concept for commanding navigation tasks. The target position is selected in a birds-eye view of the current environment. Buttons on the bottom of the content area allow to change the orientation of the navigation target.

interface for commanding a service robot using medium- and high-level command modalities.

We propose to divide the GUI in three areas as depicted in Figure 3: The *content area* serves as main interaction an visualization element of the GUI. Here, the user specifies its command interest. This interest is used to generate context- and interest-specific commands and provide these to the user in the *command area* of the GUI. By separating the specification of the command interest from the commanding itself, we make sure the user uses the robot commands carefully. The *mode selection area* allows the user to switch between different interaction modes of the *content area*. We propose to use three modes for commanding a service robot to perform manipulation, data collection, and navigation tasks.

5.1. Manipulation Mode

A first prototype of the general concept used for the Manipulation Mode mode was presented in [BLB14]. In this paper, we present a version that was enhanced for the use in space.

We use the available screen area to display the video of the robot for creating a common ground between robot and user and situational awareness for the user. A horizontal and a vertical slider, as visible in Figure 3, can be used to define a target direction of the gaze of the robot. In addition, clicking on the video sets the target accordingly. As soon as an target is defined, the respective *Look at target* command is available in the command area. When the command is clicked, the robot autonomously points its gaze to the specified target.

The objects of the world representation of the robot are semi-transparently augmented over the video. This helps the user to assess the correctness and precision of the object localization provided by the robot. To command an manipulation task, the user selects the respective object in the video by long- or double-clicking as shown in Figure 4. The chosen object is highlighted in the video and the corresponding actions are provided in the command area.

By selecting the objects in the video prior to selecting a command, the user is forced to point the gaze of the robot towards the target object. By this, the user can easily verify the localization precision and observe the later task execution. Furthermore, the number of command options visible to the user is reduced by only showing the options relevant for the current task as specified by the selected object.

For data collection, an additional button is provided on top of the video that allows the user to take snapshots of the current camera view. When the button is clicked, a high-resolution image is uploaded to the UI that can be analyzed using the Data Collection Mode.

5.2. Data Collection Mode

This mode provides access to the snapshots taken in the Manipulation Mode. An image-gallery slider shows previews and let the user scroll through the images. When the user selects a snapshot by clicking it, the corresponding image is shown full-size in the upper part of the content area as depicted in Figure 5. The user can then analyze the image and mark visible objects and/or anomalies in the image using a set of context-specific tags. The analyzed image including the tags can then be sent to a remote supervisor for further analysis.

5.3. Navigation Mode

The Navigation Mode provides utilities, which allow the user to move the robot in the environment. The target location of the robot can either be specified in the video or on a birds-eye view of the current environment by clicking at the desired position as shown in Figure 6. An overlay of the footprint of the robot is shown at the target position allowing the operator to check that the target is collision-free. The desired rotation of the robot can be changed using the corresponding buttons of the GUI. After target specification, the corresponding command is provided in the command area. When selected, the robot travels autonomously to the target avoiding collisions on the route. The progress can be monitored by using the video of the robot, that is shown on the screen of the tablet computer.

The proposed UI for tablet computers allows the user to command an autonomous service robot in an intuitive way. Special training for using the UI is not required because commanding the robot comes down to clicking at a region of interest and then letting the robot do the work. Due to the autonomous task execution, the system can be used in environments with unreliable communication links with high delays.

6. COMPARISON OF THE UI CONCEPTS

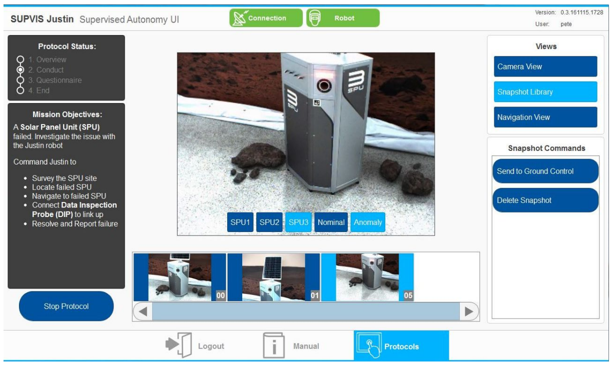
The proposed UI concepts take the restrictions and advantages of their respective interface hardware into consideration. Therefore the application domains of the concepts differ. The tablet UI concept is well suited for allowing a remote user to get situational awareness of the robot in its environment using the video provided by the camera of the robot. This is the foundation to commanding manipulation and navigation tasks that are autonomously executed by the robotic coworker. Using this interface, commanding a robot becomes a side task. It can be handled by the user, even with significant delays in the communication.

The smartphone UI concept lacks the possibility for the remote user to gain situational awareness of the environment of the robot. Therefore, it is best suited for users being in close proximity to the robot. By supplying modalities to command the robot on a low- and medium-level of autonomy, the robot can be used rather as a tool than a coworker. This puts additional workload on the user, but also allows him to resolve situations that cannot be covered by the autonomy features of the robot. Due to the direct commanding of the actuators, this UI concept is only suited for use with a reliable communication link with minimal delay.

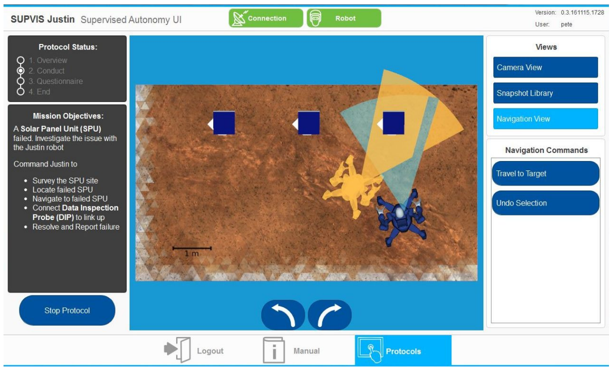
As direct commanding of robot functions could possibly damage the robot and its surrounding, the low-level autonomy command modalities provided by the smartwatch UI concept should only be used by a trained user who is familiar with the robot. In contrast, the tablet computer based UI concept can be used by an inexperienced user as well, as the robot takes care of its own safety and the safety of its environment.

The interface concepts are designed to be complementary, such that they are best used in combination. The smartwatch could be used to notify the user, that input for the robotic coworker is required. The user then uses the tablet computer interface to gain situational awareness in the environment of the robot and command the robot on a high-level of autonomy. The user can also assess if the robot was able to successfully solve the task autonomously. In case of a failure, the user can take over by

(a)



(b)



(c)

Figure 7. Screenshots of the implementation of the tablet computer UI for the METERON SUPVIS Justin ISS experiment: Manipulation Mode (a), Data Collection Mode (b), Navigation Mode (c)

commanding the robot on a low level of autonomy using the smartwatch interface. This can be achieved remotely with the video on the tablet screen, if there is a reliable communication link with low delay. It is also possible for the user to command the robot on-site. In this case, the tablet interface is not needed, leaving the user with a lightweight non-restraining UI.

First user-tests show a high acceptance rate for both UI concepts. In particular the tablet computer interface is rated to be intuitive to use. The smartwatch UI concept is preferred by the users when standing next to the robot.

7. SPACE VALIDATION OF THE TABLET UI - METERON SUPVIS JUSTIN

We aim to validate the proposed tablet computer UI for the use in space during the METERON SUPVIS Justin experiment aboard the ISS. METERON is a suite of experiments initiated by ESA with partners DLR, NASA, and Roscosmos [Sch11]. The goal of the SUPVIS Justin experiment is to explore the viability of a supervised autonomy command interface for space telerobotics [LLS+15b][LLB+17].

In the experiment, an astronaut shall take over the command of Rollin' Justin, a dexterous humanoid service robot developed at DLR [BWS+09]. Rollin' Justin shall be located in a simulated Martian solar farm located at DLR in Oberpfaffenhofen, Germany [LLS+15a]. The astronauts' objective is to command the robot to perform service-, observation-, and repair-tasks.

The proposed interface concept is embedded into the Haptics UI provided by the ESA Telerobotics & Haptics Lab [SKK+16]. The GUI consists of multiple elements for login, reading manuals, executing different experiments, and questionnaire fill-out. The execution of the experiment is done using the proposed tablet computer interface, which is extended by a *mission status* column on the left side of the GUI. This additional element is used to communicate the current mission objectives and the protocol execution status to the astronaut during the experiment. Screenshots of the resulting tablet computer application are shown in Figure 7.

The system has been tested at the Automatica Expo in 2016, where we telecommanded Rollin' Justin over a distance of ca. 50 km using a standard internet connection. A full simulation as well an astronaut training have been successfully conducted in May 2017 together with EAC in Cologne and GSOC in Oberpfaffenhofen.

The execution of the experiment on-board the ISS is currently scheduled for ISS expedition 52/53.

ACKNOWLEDGMENTS

We would like to thank the ESA Telerobotics & Haptics Lab for the excellent coopertation in METERON SUPVIS Justin. Implementing the ISS communication link would not have been possible without the support of GSOC and Col-CC. Special thanks goes to EAC and DAMEC for their support during experiment preparation, testing and astronaut training. We would also like to thank Martin Huber for implementing the prototype of the smartwatch UI.

REFERENCES

[AHY12] Syed Mohsin Abbas, Syed Hassan, and Jongwon Yun. Augmented Reality Based Teaching Pendant for Industrial Robot. In *International Conference on Control, Automation and Systems (ICCAS)*, pages 2210–2213. IEEE, 2012.

[AK16] Jong-gil Ahn and Gerard J Kim. Remote Collaboration Using a Tele-Presence Mobile Projector Robot Tele-Operated by a Smartphone. In *International Symposium on System Integration (SII)*, pages 236–241. IEEE, 2016.

[BLB14] Peter Birkenkampf, Daniel Leidner, and Christoph Borst. A Knowledge-Driven Shared Autonomy Human-Robot Interface for Tablet Computers. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 152–159. IEEE, 2014.

[BWS+09] Christoph Borst, Thomas Wimbock, Florian Schmidt, Matthias Fuchs, Bernhard Brunner, Franziska Zacharias, Paolo Robuffo Giordano, Rainer Konietschke, Wolfgang Sepp, Stefan Fuchs, et al. Rollin'Justin - Mobile Platform with Variable Base. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1597–1598. IEEE, 2009.

[CBWC15] CS Chung, J Boninger, H Wang, and RA Cooper. The Jacontrol: Development of a Smartphone Interface for the Assistive Robotic Manipulator. In *Rehabilitation Engineering and Assistive Technology Society of North America Annual Conference*, 2015.

[FMK16] Jared A Frank, Matthew Moorhead, and Vikram Kapila. Realizing Mixed-Reality Environments with Tablets for Intuitive Human-Robot Collaboration for Object Manipulation Tasks. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 302–307. IEEE, 2016.

[KHMK13] Christoph Krofitsch, Christian Hinger, Munir Merdan, and Gottfried Koppensteiner. Smartphone driven control of robots for education and research. In *International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (ROBIONETICS)*, pages 148–154. IEEE, 2013.

[LB13] Daniel Leidner and Christoph Borst. Hybrid Reasoning for Mobile Manipulation based on Object Knowledge. In *Workshop on AI-based robotics at International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013.

[LBH12] Daniel Leidner, Christoph Borst, and Gerd Hirzinger. Things Are Made for What They Are: Solving Manipulation Tasks by Using Functional Object Classes. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 429–435. IEEE, 2012.

[LBL17] Daniel Leidner, Peter Birkenkampf, and Neal Y. Lii. Context-aware Mission Control for Astronaut-Robot Collaboration. In

14th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA). ESA, 2017.

[LBLB14] Daniel Leidner, Peter Birkenkampf, Neal Y Lii, and Christoph Borst. Enhancing Supervised Autonomy for Extraterrestrial Applications by Sharing Knowledge between Humans and Robots. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, 2014.

[LCK11] Jens Lambrecht, Moritz Chemnitz, and Jörg Krüger. Control layer for multi-vendor industrial robot interaction providing integration of supervisory process control and multifunctional control units. In *Conference on Technologies for Practical Robot Applications (TePRA)*, pages 115–120. IEEE, 2011.

[LCR+12] Neal Y Lii, Zhaopeng Chen, Maximo A Roa, Annika Maier, Benedikt Pleintinger, and Christoph Borst. Toward a Task Space Framework for Gesture Commanded Telemanipulation. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 925–932. IEEE, 2012.

[LDS+14] Daniel Leidner, Alexander Dietrich, Florian Schmidt, Christoph Borst, and Alin Albu-Schäffer. Object-Centered Hybrid Reasoning for Whole-Body Mobile Manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1828–1835. IEEE, 2014.

[LLB+17] Neal Y. Lii, Daniel Leidner, Peter Birkenkampf, Benedikt Pleintinger, Bayer Bayer, and Thomas Krueger. Toward Scalable Intuitive Telecommand of Robots for Space Deployment with METERON SUPVIS Justin. In *14th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2017.

[LLS+15a] Neal Y Lii, Daniel Leidner, André Schiele, Peter Birkenkampf, Ralph Bayer, Benedikt Pleintinger, Andreas Meissner, and Andreas Balzer. Simulating an Extraterrestrial Environment for Robotic Space Exploration: The METERON SUPVIS-JUSTIN Telerobotic Experiment and the SOLEX Proving Ground. 2015.

[LLS+15b] Neal Y Lii, Daniel Leidner, André Schiele, Peter Birkenkampf, Benedikt Pleintinger, and Ralph Bayer. Command robots from orbit with supervised autonomy: An introduction to the METERON SUPVIS-Justin experiment. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 53–54. ACM, 2015.

[MBGH14] Carlos Mateo, Alberto Brunete, Ernesto Gambao, and Miguel Hernando. Hammer: An Android Based Application for End-User Industrial Robot Programming. In *International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6. IEEE, 2014.

[MS14] Simon Mayer and Gábor Sörös. User Interface Beaming–Seamless Interaction with Smart Things Using Personal Wearable Computers. In *International Conference on Wearable and Implantable Body Sensor Networks Workshops (BSN Workshops)*, pages 46–49. IEEE, 2014.

[MSB12] Sebastian Muszynski, Jörg Stückler, and Sven Behnke. Adjustable Autonomy for Mobile Teleoperation of Personal Service Robots. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 933–940. IEEE, 2012.

[Sch11] André Schiele. METERON - Validating Orbit-to-Ground Telerobotics Operations Technologies. In *11th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2011.

[SKK+16] A Schiele, T Krüger, S Kimmer, M Aiple, J Rebelo, J Smisek, E den Exter, E Mattheson, A Hernandez, and F van der Hulst. Haptics-2 - A system for bilateral control experiments from space to ground via geosynchronous satellites. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 892–897. IEEE, 2016.

[VSR+17] Valeria Villani, Lorenzo Sabattini, Giuseppe Riggio, Alessio Levratti, Cristian Secchi, and Cesare Fantuzzi. Interacting With a Mobile Robot with a Natural Infrastructure-Less Interface. *arXiv preprint arXiv:1704.00905*, 2017.