

Using Augmented Reality to Improve Usability of the User Interface for Driving a Telepresence Robot

Giovanni Mosiello^{12*}, Andrey Kiselev^{2†}, Amy Loutfi^{2‡}

¹ Università degli Studi Roma Tre,
Via Ostiense 159, 00154 Rome, Italy

² Örebro University,
Fakultetsgatan 1, 70182 Örebro, Sweden

Received 18-09-2013

Accepted 12-12-2013

Abstract

Mobile Robotic Telepresence (MRP) helps people to communicate in natural ways despite being physically located in different parts of the world. User interfaces of such systems are as critical as the design and functionality of the robot itself for creating conditions for natural interaction. This article presents an exploratory study analysing different robot teleoperation interfaces. The goals of this paper are to investigate the possible effect of using augmented reality as the means to drive a robot, to identify key factors of the user interface in order to improve the user experience through a driving interface, and to minimize interface familiarization time for non-experienced users. The study involved 23 participants whose robot driving attempts via different user interfaces were analysed. The results show that a user interface with an augmented reality interface resulted in better driving experience.

Keywords

Human-Computer Interaction · Mobile Robotic Telepresence · Augmented Reality · Usability · User Interface Design · Domestic Robotics · Giraff

© 2013 Giovanni Mosiello et al., licensee Versita Sp. z o. o.

This work is licensed under the [Creative Commons Attribution-NonCommercial-NoDerivs license](#), which means that the text may be used for non-commercial purposes, provided credit is given to the author.

1. Introduction

Mobile Robotic Telepresence (MRP) helps people to communicate in natural ways despite being physically located in different parts of the world. This is achieved by allowing users connecting from remote locations to not only convey vocal and visual information, but also use gestures and poses to the extent which particular MRP robots allow. An important aspect of MRP is social interaction where a robot equipped with standard videoconferencing technologies enables audio and video communication. Many MRP devices [12] are emerging on the market and have been advocated for applications within home and office environments. Likewise several research works have examined various aspects of MRP systems such as social acceptance [1] or quality of interaction [11].

Like other telepresence robots used in industrial [20], scientific [7, 18], rescue [3], and medical [9, 19] applications MRP systems for social interaction are now starting to enter consumer markets. One particular challenge this market faces is the variety of novice users, their skills and experiences. "Teleoperation is indirect process" [6] and this requires users to be able to learn how his/her interaction with the user interface (UI) affects actual robot behaviour. This ability can be trained, but the acceptable amount of necessary training for consumer applications is very strict. This makes the interface between novice users and the robot especially important. This is particularly challenging in modern

MRP systems as steering the robot is not the primary task and it should not detract from the quality of social interaction. It has been shown in previous studies [13] that efficiency and safety of operation might be limited by interface design. Further studies on interfaces such as [10] have also shown that an individuals' workload can vary and has a direct relation on driving performance. Therefore it is important to consider how the UI could be designed so that the task of driving does not create a heavy workload for novice users.

In this paper we examine the possibility of augmenting the visitor (or pilot) interface to enable improved spatial perception with the aim of improving the sense of presence [2, 16]. It has been previously shown that augmented reality can be "a useful tool to help non-technical operators to drive mobile robots" [15]. In our study the interface is augmented by a visual cue in the pilot interface which projects the dimensions of the robot onto the driving surface. This projection assists drivers in determining how much room there is to navigate the robot, leading to fewer collisions and better performance, especially for the users whose are not used to playing video games (novice users). The purpose of this research is to measure the improvement of performance and to identify the key factors in making the user interface "friendly" and effective.

Experimental validations are made by comparing different variations of the interface in order to ensure that performance is not dependent only upon an aesthetic quality of the interface. Further, the performance of both novice and expert users (users whose use to play video games) is investigated.

The paper is organized as follows. Section 2 provides a system overview and shows various interfaces used in this study. Section 3 outlines the method for collecting data about users' behaviour, and Section 4 details the results. The paper is concluded by Section 5.

*E-mail: gio.mosiello@gmail.com

†E-mail: andrey.kiselev@oru.se

‡E-mail: amy.loutfi@oru.se



Figure 1. Giraff robot.

2. System Overview

The Giraff Robotic Telepresence system [8] is used for the current experiment. It includes the Giraff robot (see Fig. 1) and Pilot interfaces, interconnected by the Giraff Sentry service. The Giraff system allows “embodied visits”¹ from any remote location² with sufficient Internet connection. Any MS Windows-based PC can be used. The Giraff Sentry service verifies user credentials and provides information about robots’ availability and status to allowed users.

2.1. Pilot Application

The pilot application allows the remote user to establish connection with a robot, steer the robot and communicate with local users. Three versions of the Giraff Pilot application are compared side-by-side in this study. These are Giraff Pilot 1.4 (see Fig. 2), Giraff Pilot 2.0a (see Fig. 3), and Giraff Pilot 2.0b (see Fig. 4). Giraff Pilot software can be installed on any MS Windows-based PC with Java Runtime Environment version 1.6 or higher.

The robot is steered using any standard pointing device such as a mouse or a touch pad. This is similar in all Pilot versions. To drive the robot, the user has to move the pointer into the local video stream area and click at the desired location, holding down the left mouse but-

¹ By “embodied visits” we mean visiting a local environment from a remote location, while being embodied into any kind of physical device or vehicle.

² The following definitions of environments are used in this paper: the *local environment* is the one where the robot is physically located, the *remote environment* is any environment from which the Pilot application is used to control the robot.

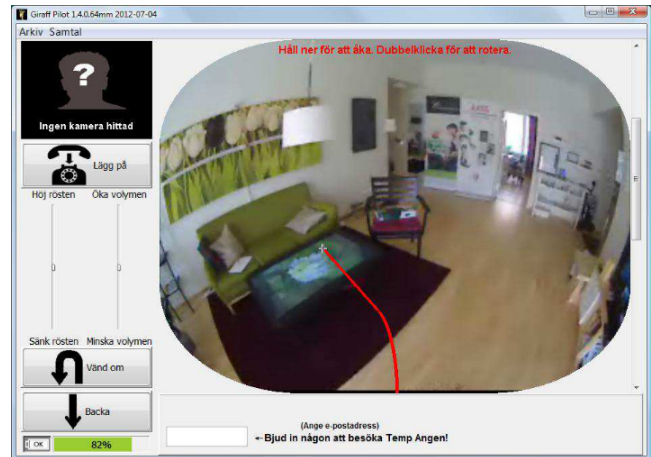


Figure 2. Giraff Pilot 1.4

ton to make the robot move. The further the point is from the robot the faster the robot will move. The robot keeps moving as long as the left mouse button is kept pressed.

All versions of Giraff Pilot have similar layouts. The greater portion of the window is occupied by the video stream from the robot's camera. This is the main application panel, and there is also a control panel to the left of the window.

2.1.1. Pilot 1.4

Giraff Pilot 1.4's control panel incorporates remote user video (to allow pilots to see themselves), call control button, audio controls, “rotate” button, “back-up” button and battery status informer. It has a standard Windows look and feel.

2.1.2. Pilot 2.0a

Pilot 2.0's panel is slightly different: robot steering controllers and a battery status informer were dropped, but instead there are three buttons to adjust camera, video and audio settings. The greatest visual difference between Giraff Pilot 1.4 and both versions of Pilot 2.0 is that the latter has a Nimbus look and feel [17]. This allows better aesthetic quality of the application and theme alignment to Giraff's corporate color scheme.

Pilots 1.4 and 2.0a use a line to show the trajectory which the robot is supposed to follow. This line is red while the robot is still and becomes green as soon as the robot starts to move towards the destination.

2.1.3. Pilot 2.0b

The aiming method for Pilot 2.0b is different from the other Pilot versions: the path estimation line has been changed, with a target which projects the dimensions of the robot to the driving surface using perspective transformation according to the position of the camera (see Fig. 4). The target has three different modes:

1. **blinking blue** — idle (the robot is still);

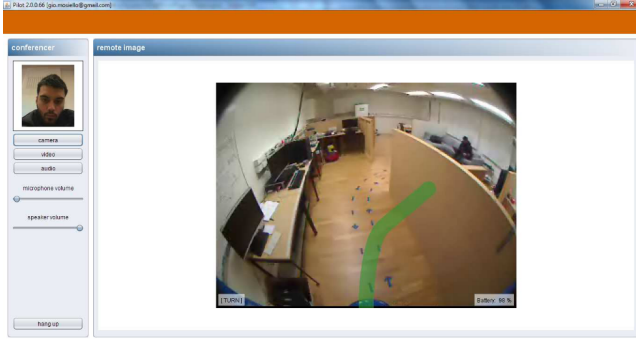


Figure 3. Giraff Pilot 2.0a

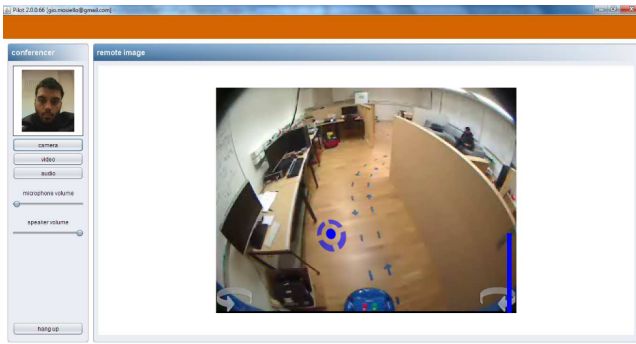


Figure 4. Giraff Pilot 2.0b

2. **blinking red** — the desired position is not reachable³;
3. **rotating blue** — the robot is moving.

There are two buttons in the local video area to allow the user to perform a 180° rotation (clockwise and counter-clockwise). Additionally, a speedometer is placed on the right side of the video stream.

The size (diameter) of the target corresponds to the dimensions of the robot (its wheelbase). The actual target is an image, which is transformed according to several parameters. The exact shape of the target is calculated using the position of the pointer on screen (x and y coordinates relative to the center of the robot's wheelbase), the actual position (height and tilt) of the camera relative to the floor (calculated using the known robot's physical parameters), and a camera lens distortion model. This transformation is recalculated every time the pointer is moved or the robot's head tilt is changed. Thus the target always appears augmenting the local environment. Additionally, the target is slightly rotated when being moved to the left or right side of the robot to give a better spatial effect when mouse pointer is moved.

The transformation is done in three steps. First, the actual orientation of the camera and position of its focal point relative to the ground plane are calculated according to the robot's dimensions (see Fig. 5) using

³ Those set of positions which, because the API implementation, cannot be reached directly by the robot. The robot will not move towards the target.

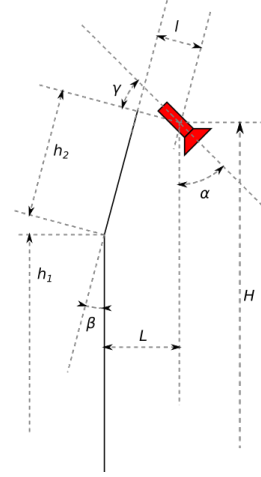


Figure 5. Giraff Camera Position Calculations

Eq. 1 and Eq. 2:

$$H = h_1 + \sin \left(90^\circ - \beta - \sin^{-1} \left(\frac{l}{\sqrt{h_2^2 + l^2}} \right) \right) \sqrt{h_2^2 + l^2}, \quad (1)$$

where H is the height of the camera's focal point from the ground.

$$\alpha = \beta + \gamma, \quad (2)$$

where α is the camera's actual angle, β – head tilt angle, and γ – camera installation angle relative to the head.

At the next step we use the standard perspective transformation matrix (see Eq. 3, where *aspect* is an aspect ratio of the projection plane, *fovy* is the lens' vertical field of view, zN and zF – near and far clipping planes respectively) to project the target onto the floor in a position which is known from H and α , assuming that camera rotation around the z – axis is always 0.

$$\begin{pmatrix} \frac{1}{\text{aspect} \cdot \tan(\frac{\text{fovy}}{2})} & 0 & 0 & 0 \\ 0 & \frac{1}{\tan(\frac{\text{fovy}}{2})} & 0 & 0 \\ 0 & 0 & \frac{-zN - zF}{zN - zF} & \frac{2 \cdot zF \cdot zN}{zN - zF} \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad (3)$$

In our case $\text{aspect} = \frac{320}{240}$, $\text{fovy} = 100^\circ$.

Finally, we apply barrel distortion to the target using Eq. 4:

$$\begin{aligned} x' &= \frac{w}{2} - w d_x (\alpha_x + \beta_x |d_y| + \gamma_x d_y^2 + \delta_x d_y^4); \\ y' &= \frac{h}{2} - h d_y (\alpha_y + \beta_y |d_x| + \gamma_y d_x^2 + \delta_y d_x^4); \\ d_x &= \frac{\frac{w}{2} - x}{w}; \\ d_y &= \frac{\frac{h}{2} - y}{h} \end{aligned} \quad (4)$$

where w and h are width and height of the image respectively, x, y and x', y' source and target coordinates of each pixel, $\alpha_x, \beta_x, \gamma_x, \delta_x, \alpha_y, \beta_y, \gamma_y, \delta_y$ are correction coefficients, found empirically.

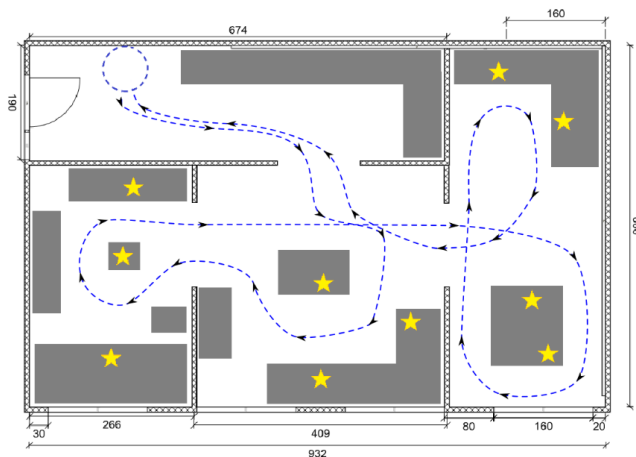


Figure 6. Experiment path (dashed line) and checkpoints (stars)

3. Method

The goal of the series of experiments is to collect data about the users' behaviour in using a particular control interface. The scope is to measure performances in terms of time to accomplish a typical task and using a questionnaire to assess users' attitudes.

Subjects are asked to undock the robot from the charging station, drive through the environment following a path (dashed line) on the floor (see Fig. 6), find a secret code in one of ten possible locations, and dock the robot back to the charging station. Subjects' performance in terms of time to complete the task and number of collisions is calculated during the driving session. All subjects connect to the robot from a remote location to avoid any prior exposure to the local environment. There are ten possible locations where a secret code can be found, but its exact location is not known until it is found.

Therefore, the detailed procedure for each subject is as follows:

1. undock the robot from the charging station;
2. drive through the experimental environment following the dashed line;
3. check all the possible locations to find a code;
4. write down the code if found;
5. proceed to remaining locations following the dashed line;
6. dock the robot back to the charging station.

When the subject has sufficiently docked the Giraff the task is accomplished and the time spent between undocking and docking appears on-screen. The user has to write down the time and answer a general questionnaire about driving experience. The best performing participants obtain a certificate called "Giraff Driving License". The requirements for obtaining the license are:

- (time spent) < (best time) * 3;
- (collisions) < 4

During the entire experiment, the screen is recorded for future analysis.

3.1. Subjects

Twenty three subjects participate in the experiment (17 males and 6 females), the average age was 22.26 years (SD 1.69). Subjects were divided in three groups (7 subjects used Pilot 1.4, 7 uses Pilot 2.0a and 9 uses Pilot 2.0b), each group tested only one version of Giraff Pilot.

None of the subjects had prior experience with this kind of experiment. The only prior experiences were about gaming, those subjects were split into the three groups (information about previous experiences in gaming came from the questionnaire).

3.2. Questionnaire

The questionnaire focused on evaluating the overall reactions to the software, the learning (how the user learned to use software) and the system's capabilities. The first part of the questionnaire contained questions to profile the user and boxes to enter the code and the time spent in completing the path. The second part of the questionnaire contained 29 questions organized into three groups: performance, aesthetics and engagement. Seven-item Likert scales were used for each question. The questionnaire was organized according to QUIS directions [4]. The list of questions was as follows:

A Performance

- A.1 The first approach with Giraff;
- A.2 The task accomplishment;
- A.3 The docking Procedure;
- A.4 Following the path;
- A.5 Estimating spatial dimensions;
- A.6 Searching for the targets;
- A.7 Driving Giraff through doors;
- A.8 Using the Giraff is practical;
- A.9 Giraff takes a lot of effort to use properly;
- A.10 Giraff is easy to use;
- A.11 Drive Giraff is intuitive;
- A.12 Using Giraff to complete the task is effective;
- A.13 Giraff has good performance.

B Aesthetics

- B.1 Pilot is clean;
- B.2 Has good aesthetics;
- B.3 Is original;
- B.4 Is colored;
- B.5 Has beautiful shapes;
- B.6 Is clear.

C Engagement

- C.1 I had a lot of fun;
- C.2 I felt everything was under control;
- C.3 Was exciting;
- C.4 I felt frustrated;
- C.5 Was technical;
- C.6 I felt pleasure;
- C.7 I want to use Giraff in future;
- C.8 Giraff has a high-quality user interface;
- C.9 Giraff is sophisticated;
- C.10 Giraff was stressful.

Questions from A.1 to A.7 utilize "very difficult — very easy" scale, while others use options from "strongly disagree" to "strongly agree". Question A.13 is a control question designedly biased and is not considered in conclusions. Questions A.2, A.6, B.1, B.6, C.4, C.6 and C.10 are control questions included to check the reliability.



Figure 7. Example of experiment path (detail)

3.3. Environment

Giraff was placed in the PEIS Home environment⁴. Obstacles have been added in order to recreate real home conditions. A path has been drawn on the floor using a bright blue dashed line with arrows (see Fig. 7). The path contains a number of typical curves along with a long straight stretch (around 5m) to test the users' behaviour in situations where high speed is allowed (see Fig. 6).

4. Results

Figures 9–17 show Giraff Pilot 1.4, Giraff Pilot 2.0a, and Giraff Pilot 2.0b results, with data grouped according to users' experience in computer games. The reason for distinguishing users based on their gaming experience is that experienced gamers look at the driving interfaces using their experience in gaming, so they are more sensitive in evaluating the quality of the elements of the software.

The overall task completion rate in this study was 74%. The success conditions for each participant are to 1) complete the path and dock the robot to the charging station and 2) to read the secret code correctly. The number of collisions was not taken into account for calculating the task completion rate as it was not critical for this measurement.

4.1. Performance

Performance measurement was conducted by monitoring the time taken to complete the path (time between undocking and docking the Giraff robot). All questionnaire results were reliable, none were eliminated (according to the answers to the control questions). Participants who saw and reported the control code took, on average, 5.4s (SD 3.2s) to stop the robot and orient Giraff's camera to the code, and 5.9s (SD 2.2s) to write down the code on the questionnaire. Thus, a penalty

⁴ The PEIS Home is an experimental environment looking like a typical bachelor apartment of about 55 square meters. It consists of a living room, a bedroom, and a small kitchen. The walls have been constructed to be 1.40 meters high, so that observers can get a bird's eye view of the entire apartment



Figure 8. Collision Policy. Collisions in yellow areas are counted if the contact lasts longer than 3 seconds; collisions in red are counted immediately.

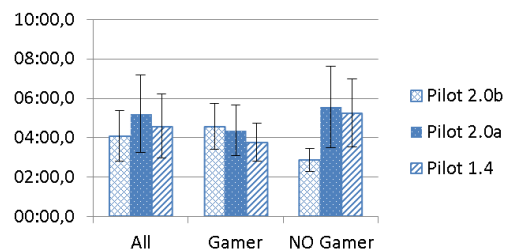


Figure 9. Performance in terms of time (means and SDs for different groups of users).

of 15s was applied to the participants who did not find the code, in order to normalize their times.

All subjects who collided with obstacles had to spend more time completing the path (backing up and then continuing), therefore no penalty was applied for collisions.

In order to get information about the ability of the software to help to avoid collisions, the latter have been counted using the following parameters (see Fig. 8):

- if Giraff collides in red area → collisions +1;
- if Giraff collides in orange area **and** contact lasts longer than 3s → collisions +1.

Looking at the performance results of subjects who are not used to playing video games, we can see a significant improvement in terms of the time they spent completing the path. At the same time, the occurrence of collisions tends to decrease using both Pilot 2.0a or 2.0b. For those who are used to playing video games there is no significant difference, but for non-gamers the improvement is more evident.

4.2. Results of the Questionnaire

The questionnaire results are a qualitative analysis based on the users' perception.

Spatial perception is one of the main targets of this study. Users' subjective reports show that the best spatial perception can be achieved with the Pilot 2.0b (using driving target), while the worst result has been shown by the Pilot 1.4 (see Fig. 11).

Pilot 2.0b outperforms other versions in terms of effort needed to steer the robot (see Fig. 12). This result is especially strong for non-gamers, who have no prior exposure to teleoperation.

Pilot 2.0a makes it easier to follow the path, compared to other interfaces. This is more evident for gamers (see Fig. 13), but navigation

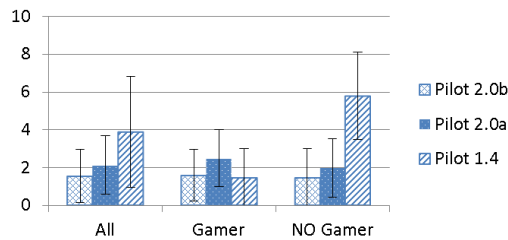


Figure 10. Performance in terms of collisions (means and SDs for different groups of users).

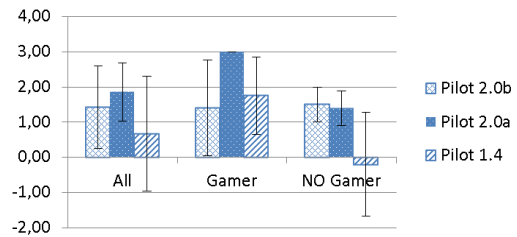


Figure 13. Following path

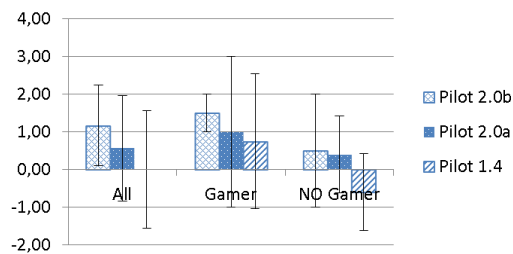


Figure 11. Spatial perception

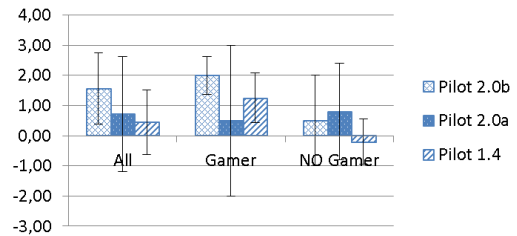


Figure 14. Pass through narrow passages

through narrow passages (e.g. doorways) is simpler using Pilot 2.0b (see Fig. 14).

Pilot 2.0a and 2.0b both tend to be more intuitive than Pilot 1.4. The highest intuitiveness is achieved by the Pilot 2.0a thanks to strong opinions of gamers.

Pilot 2.0a tends to be easier to use than others. An interesting result is that gamers prefer the Pilot 2.0b, but non-gamers find Pilot 1.4 to be easier to use (see Fig. 15).

Analysing the visual pleasantness of the interface, Pilot 2.0b offers a better user experience according to the users' opinion (see Fig. 16 and Fig. 17). This result is confirmed by the users' perception of a higher quality of Pilot 2.0b. The result about the "clearness" is also in line with the previous results: Pilot 2.0b looks clearer than the others.

When it comes to users' attitudes, users are inclined to be more stressed using Pilot 2.0b, but for non-gamers this difference is insignificant. The aiming method (line or target) makes a significant differ-

ence for different user groups: gamers prefer the driving line, while non-gamers prefer the target.

4.3. Additional comments and observations

Several observations emerged during the experiment. The lag between mouse click and Giraff movement makes driving difficult; novice users performed many double clicks before understanding how to drive properly. The majority of the participants had problems moving backwards, and only one subject used the **move backwards** button in Pilot 1.4 (see Fig. 2). Seven users did not understand the function of the speedometer in Pilot 2.0b. A common factor was the difficulty in understanding how the robot would move according to the user commands, especially for slow movements (getting to destinations close to the robot). The experiment was biased by the height of walls inside the

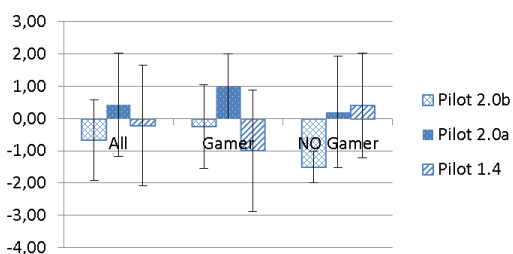


Figure 12. Effort needed

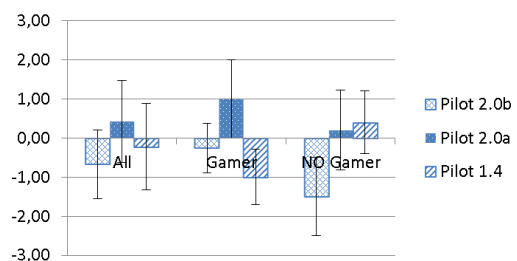


Figure 15. Easiness

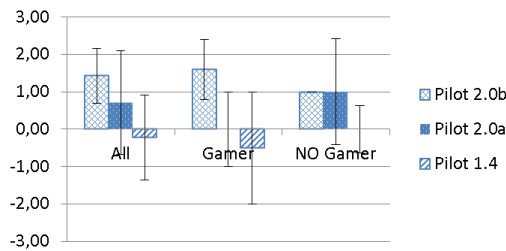


Figure 16. Aesthetics

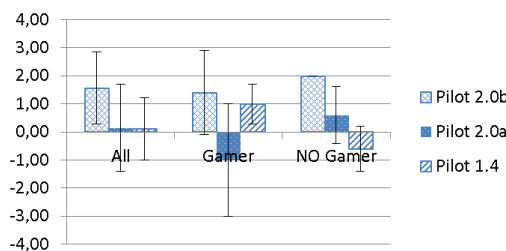


Figure 17. Quality Perception

PEIS home: 1.4m allows users to see “behind the walls”. It helps them to fix the trajectory before passing a doorway⁵.

5. Discussions and Future Works

Pilot 1.4 and 2.0a both show a path on the screen but in terms of intuitiveness they have different results. This can be attributed to the different thickness of the path shown on-screen, and different aesthetics of the entire application. This also explains the users’ feedback about spatial perception: both Pilot 2.0a and 2.0b allow better spatial perception and simplify navigation in narrow spaces. The thickness of the path in Pilot 2.0a gives the users an idea of the space which is going to be occupied by Giraff; in Pilot 2.0b this space is exactly estimated.

As shown in 4.3, Pilot 2.0a allows users to more easily follow the path and change direction while the robot is moving.

Pilot 2.0b is less intuitive than 2.0a. This was expected because the prospective path is not shown in Pilot 2.0b and users have difficulties in understanding what movement the robot is going to perform, especially for precise manipulations.

One of the problems, which users normally encounter, is to infer the robot size in order to avoid collisions. Projecting the robot’s dimensions onto the prospective destination is only helpful to users for short movements [14]. The best solution, and one direction of future work, seems to combine both aiming techniques (path and target) to offer cues for better spatial perception and trajectory estimation at the same time.

⁵ Previous experiments in a different environment with real walls shows worse performance in previsualizing the trajectory

In general, users tend to ignore movement controls outside the video stream area, as they interpret the buttons as not relating to controlling the robot. For this reason all the robot control buttons have to reside in the video stream area. In addition buttons should contain a clear icon showing their function (e.g. **turn** button, see Fig. 3 and Fig. 4), otherwise users tend to ignore them (see 4.3).

The time needed to perform the path does not vary using different Pilot versions (see Fig. 9), but after analysis of recorded video (see 4.3), we can infer that using key factors of both Pilot 2.0a and Pilot 2.0b, it will be possible to reduce the time needed for *command understanding* and *size understanding* phases; the time needed for the entire task will be reduced.

As shown in Fig. 10, the occurrence of collisions decreases when showing the projection of the robot into the remote environment. This means that having an on-screen projection of the space the robot is going to fill helps the user to avoid collisions.

Summarizing the conclusions, an effective and efficient driving interface for driving a telepresence robot should:

- ‘ project the final position of the robot;
- ‘ show the path the robot is going to perform (projected onto the driving surface with respect to robot’s camera orientation);
- ‘ offer alternative ways to steer the robot;
- ‘ have any controls related to steering a robot inside the video stream area;
- ‘ clearly show information about the robot status (e.g. speed and head tilt);
- ‘ reduce any control latency between the mouse click and the robot’s movements.

This conclusions are especially important when designing interfaces for novice users, who have no knowledge or experience about robotic telepresence.

The conducted experiment also shows extremely diverse preferences of different participants to different teleoperation interfaces, especially for those users who have experience with video games which nowadays can offer high levels of immersion. This certainly must be taken into account when designing teleoperation interfaces for MRP systems.

6. Summary

In this study we compared three different versions of the pilot interface of the Giraff Mobile Robotic Telepresence system. One of the interfaces is designed to use visual cues to enhance users spatial perception and give better feeling of dimensions and distances when driving a telepresence robot. In the series of experiments we confirmed that visual cues can improve performance, especially for those users, who have no prior experience with teleoperation and video games. At the same time, this study suggests that combining different aiming methods (e.g. showing the robot’s tentative moving path and augmenting local environment) can offer a much better user experience, leading by turn to smoother communication between remote and local users.

Acknowledgements

This work has been made possible thanks to strong support from the Centre of Applied Autonomous Sensor Systems, Örebro University, Sweden and Ambient Assisted Living Joint Program – ExCITE Project [5] (AAL-2009-2-125).

References

- [1] Jenay M Beer and Leila Takayama. Mobile Remote Presence Systems for Older Adults : Acceptance , Benefits , and Concerns. In *Proceedings of the 6th international conference on Human-robot interaction HRI 11*, HRI '11, pages 19–26. ACM, 2011.
- [2] Oliver Bimber and Ramesh Raskar. *Spatial Augmented Reality Merging Real and Virtual Worlds*, volume 6. A K Peters, Ltd., 2005.
- [3] Kristoffer Cavallin and Peter Svensson. *Semi-Autonomous, Teleoperated Search and Rescue Robot*. PhD thesis, 2009.
- [4] John P Chin, Virginia A Diehl, and Kent L Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In Elliot Soloway, Douglas Frye, and Sylvia B Sheppard, editors, *Learning*, volume 218 of *CHI '88*, pages 213–218. ACM New York, NY, USA, ACM, 1988.
- [5] S. Coradeschi, A. Kristoffersson, A. Loutfi, S. Von Rump, A. Cesta, G. Cortellessa, and J. Gonzalez. Towards a Methodology for Longitudinal Evaluation of Social Robotic Telepresence for Elderly. In *Proceedings of the HRI 2011 Workshop on Social Robotic Telepresence*, pages 1–7, 2011.
- [6] Anca D Dragan, Siddhartha Siddhartha Srinivasa, and Kenton Lee. Teleoperation with Intelligent and Customizable Interfaces. *Journal of Human-Robot Interaction*, 2(2):33–57, June 2013.
- [7] Terrence Fong and C Thorpe. Vehicle teleoperation interfaces. *Autonomous robots*, pages 9–18, 2001.
- [8] Giraff Technologies AB. Giraff Technologies AB, 2013.
- [9] H Iseki. Computer Assisted Neurosurgery. *International Journal of Computer Assisted Radiology and Surgery*, 1(S1):293–310, May 2006.
- [10] Andrey Kiselev and Amy Loutfi. Using a Mental Workload Index as a Measure of Usability of a User Interface for Social Robotic Telepresence. *Workshop in Social Robotics Telepresence*, 2012.
- [11] Annica Kristoffersson, Silvia Coradeschi, KS Eklundh, and Amy Loutfi. Towards measuring quality of interaction in mobile robotic telepresence using sociometric badges. *Panadyn Journal of Behavioral Robotics*, 2013.
- [12] Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi. A Review of Mobile Robotic Telepresence. *Advances in Human-Computer Interaction*, 2013:1–17, 2013.
- [13] D Labonte, F Michaud, P Boissy, H Corriveau, R Cloutier, and M A Roux. A Pilot Study on Teleoperated Mobile Robots in Home Environments, 2006.
- [14] Daniel Labonte, Patrick Boissy, and François Michaud. Comparative analysis of 3-D robot teleoperation interfaces with novice users. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 40(5):1331–42, October 2010.
- [15] FJ Rodríguez Lera. Augmented reality to improve teleoperation of mobile robots. pages 1–6, 2011.
- [16] F Michaud, P Boissy, and D Labonte. Telepresence Robot for Home Care Assistance. *AAAI Spring Symposium*, 2007.
- [17] Oracle. Nimbus Look and Feel, 2013.
- [18] RA Peters. Robonaut task learning through teleoperation. In *ICRA'03*, pages 2806–2811, 2003.
- [19] Giuseppe Riva, Francesca Morganti, and Marco Villamira. Immersive Virtual Telepresence: virtual reality meets eHealth. *Studies in health technology and informatics*, 99:255–62, January 2004.
- [20] Jussi Suomela and Aarne Halme. Tele-Existence Techniques of Heavy Work Vehicles. pages 29–38, 2001.