Influence of Human-Computer Interface Elements on Performance of Teleoperated Mobile Robot

Stanko Kružić[1], Josip Musić[1], Ivo Stančić[1]

[1]University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture
E-mail: {skruzic,jmusic,istancic}@fesb.hr

***Abstract*—Mobile robots are becoming ubiquitous, with applications which usually include a degree of autonomy. However, due to uncertain and dynamic nature of operational environment, algorithms for autonomous operation might fail. In order to assist the robot, the human operator might need to take control over the robot from remote location. In order to efficiently and safely teleoperate the robot, the operator has to have high degree of situational awareness. This can be achieved with appropriate human-computer interface (HCI), so that the remote environment model constructed with sensor data is presented at appropriate time, and that robot commands can be issued intuitively and easily. In the research, influence of HCI elements on performance of teleoperated mobile robot was studied for several tasks and with several HCI setups. The user study was performed, in which accuracy and speed of completion of given tasks were measured on a real robot. Statistical analysis was performed in order to identify possible setup dependencies. It showed that, in majority of analysed cases and based on introduced metrics, there is no significant difference between the setups, and between the visual control and teleoperation. Finally, conclusions were drawn with emphasis on benefits of information technology in particular case.**

I. Introduction

Mobile robots are becoming more popular for various applications in everyday life [1], [2], [3]. The robots usually have a certain degree of autonomy while performing their tasks. However, occasionally it is necessary that the operator takes control over the robot, usually from remote location, either because robot's algorithms for autonomous operation failed or in order to complete a task that robot cannot complete during autonomous operation.

For efficient teleoperation of the robot, the operator must be well aware of robot's state and state of its environment. This is often referred to as situational awareness [4]. Situational awareness is understanding of the conditions of dynamic environment and is defined as "*The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future.*" [5]. When interacting with a robot, it is also required that the operator is aware of the consequences of the robot's actions both for the robot and for the operational environment. Hence, high degree of situational awareness is of great importance because it is a prerequisite for the efficient and safe teleoperation. This in turn can be achieved with appropriate and efficient design of HCI which should present all necessary information about the robot, it's state and the state of the environment. The interface should be designed in a way that it offloads some of the workload from the operator and makes robot control easy and intuitive.

For the purpose of this research, safe navigation is defined as one with minimal (preferably zero) collisions with objects from the robot's operational environment. This is not always a trivial task due to dynamic and uncertain nature of the robot's environment, which is made more difficult by implementation of teleoperation control. Given only live video information from a camera mounted on the robot, operator is usually not sufficiently aware of the state of the robot's environment, especially robot's distance to obstacles (lack of depth perception), and possible nearby obstacles (camera's limited field-of-view). Therefore, some additional data must be provided to improve the operator's awareness. This can be achieved by using different sensing modalities like infrared, ultrasound or laser-based proximity sensors, RGB-D cameras, panospheric cameras, or usage of multiple cameras mounted on the robot. Additional information that is often included in navigation tasks, which provides limited amount of data, is the environment map. It can be a helpful aid, since the operator may determine robot's pose within environment at any time during navigation, as well as provide approximate information about proximity of (static) obstacles, and can be used as a reference.

The manner in which information from sensing devices is presented to the operator also affects performance of teleoperation [6]. Graphical interfaces containing multiple information fused in one frame (a form of augmented reality) were shown to be more efficient than arrangement when they were presented in multiple frames, side-by-side. Additional element that can influence operator's performance in teleoperation tasks is choice of HCI, which is considered as means for communication and interaction between the human and computer [7]. Choosing efficient HCI is a challenging task and may not be unique for different operators, because it may depend on operators' level of training [8]. Studies showed that virtual reality (VR) approach to teleoperation of robots is possible as a way to enhance both display and situation awareness [9]. Additional benefit of virtual reality is that safe training of operator in teleoperation can be achieved.

Another important variable that needs to be addressed is time (network) delay. It is well known that teleoperation is affected by latency, that may have a significant impact on teleoperation performance [10]. High time delay can make teleoperation impossible regardless of quality of HCI, while variable time delay (which is frequent in practical applications) can contribute further to degradation of performance in teleoperation. The sources of delay are various, and include both network delays and processing and sensing delays. The impact of time delay on teleoperation performance is not studied in this paper, but is only recorded and reported. This means that it changed freely, but due to dedicated network equipment used in the experiment is kept as low as possible.

Teleoperation of a mobile robots has been extensively studied topic and various and innovative approaches have been proposed recently. In [11], various aspects of newly proposed ecological 3D interfaces are compared to traditional 2D interfaces, while in [12] different control interfaces (including virtual reality based ones) are compared in order to assess impact on operator's performance. In [13], a haptic interface is proposed for teleoperation of mobile robots. However, to our best knowledge, majority of research on teleoperation was done in simulations [6], [11], [10]. In the paper we present a study of teleoperation that was performed on a real mobile robot, and draw conclusions based on it.

The rest of the paper is structured as follows. In Section 2 experimental design and setup are explained and used analysis tools are briefly presented. Section 3 presents obtained results as well as associated discussion, while in Section 4 conclusions are drawn and possible future research directions discussed.

II. Materials and Methods

A. Experimental Design

The experiment was designed as a between-subject user study where each participant had to complete set of four tasks using one of four setup designs. One variable was input method which had two levels: gamepad (joystick) and steering wheel. Second variable was graphical interface elements, again with two levels: with and without depth information. This resulted in 2x2 design setup. Besides depth information, both graphical interfaces included live video from robot-mounted camera and map of the location with robot model localized within it.

The user study was performed with 32 test subjects, 8 in each of four setups: gamepad + GUI (Graphical User Interface) without depth, gamepad + GUI with depth, steering wheel + GUI without depth and steering wheel + GUI with depth. Participants were all volunteers recruited from Faculty staff and students which gave informed consent for participation in the study. Summary of participants' demographics are presented in Table I.

The experiments were performed using *Turtlebot 2*, differential drive mobile robot, depicted in Fig. 1. Developed software support was based on Robot Operating System (ROS) [14]. The robot was equipped with a laptop, running Ubuntu 14.04 with ROS Indigo (which was used as a ROS master), live camera and 3D Kinect sensor with 640 px $\times$ 480 px resolution and 30 fps for both camera and depth streams.

TABLE I
PARTICIPANTS' DEMOGRAPHIC DATA

(F)emale and (M)ale.

Setup	No.	F/M	Age [years]	
			μ	σ
Gamepad + GUI without depth	8	5/3	30.63	5.04
Gamepad + GUI with depth	8	2/6	31.63	6.23
Wheel + GUI without depth	8	1/7	33.13	10.08
Wheel + GUI with depth	8	4/4	32.38	6.28
OVERALL	32	12/18	31.94	6.87

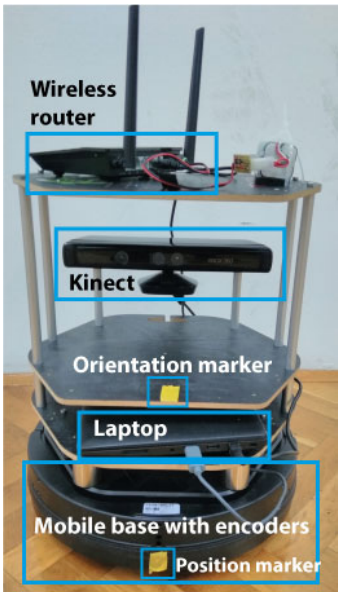


Fig. 1. An overview of Turtlebot 2 robot used in the experiments

The communication between the robot and the PC on which experiment was conducted was based on dedicated IEEE 802.11n wireless network. On robot's side, Asus RT-N12+ wireless router was used, while on PC side, running Ubuntu 16.04 and ROS Kinetic, TP-Link TP-WN722N USB wireless antenna was used. This setup ensured robot range of about 25 m radius (in non-ideal, out-of-line-of-sight conditions). The setup is depicted in Fig. 1. Robot's base ROS node (used for robot control) and Kinect driver for capturing both live camera and depth data were running on robot's laptop, while control interface driver, map (localization) and depth information interpreter (if used) were running on a PC.

The map of the environment (Fig. 2) in which the experiments were conducted was also provided to the test subjects, with robot's location visible on it in the real time. The map itself was built prior to the experiments using LIDAR sensor mounted on another (in-house built) robot and ROS package *gmapping*, which is based on [15]. More advanced robot was used for mapping due to better odometry and depth sensors, which resulted in more precise environment map.

All test subjects were given 10 minutes before the experiments to get familiar with the system and to try controlling the robot (both visually and in teleoperation mode).

Fig. 2. Map of the entire experimental environment. Please note areas used for particular task. Tasks T1 and T2 were conducted in the green marked area, T3 in the red marked area, and T4 in the blue marked area.

(a) Gamepad (b) Steering wheel

Fig. 3. Control interfaces used in the experiment

1) Control Interfaces: All control interfaces used in the experiment were programmed so that the operator could use holonomic or differential control mode and could easily switch between them (if needed). Also, all interfaces have keys for both smooth and emergency stop. Control interfaces that were used in the experiment were MS Industrial Console 6in1 wireless gamepad which communicates on 2.4 GHz frequency with 6 m range, and Trust GXT 570 gaming steering wheel (which was used without floor pedals).

Both control interfaces (gamepad and steering wheel) used the same command interpreter that generated robot's velocity commands based on input commands given by the test subject. The maximum velocity of the robot was set to 0.45 m/s for linear motion and 1 rad/s for angular motion, with constraint that the velocity of each robot wheel cannot exceed 0.5 m/s in case of combined linear and angular motion. This

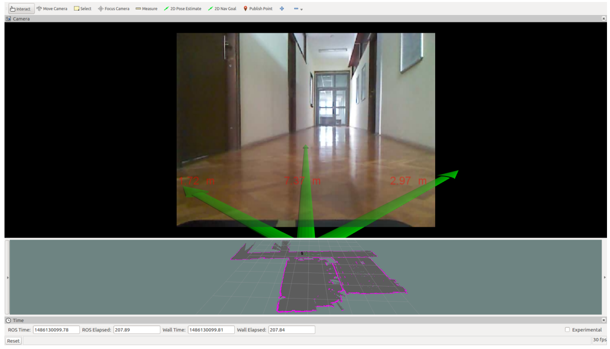


Fig. 4. GUI with depth information (green arrows) used in the experiments. When not using depth information, GUI is identical with exception of arrows.

configuration enabled cornering while driving at maximum allowed linear velocity.

2) Graphical User Interface: The intuitive and efficient GUI is one of the key factors for efficient and safe navigation of the robot from remote location. Our aim was to assess if performance of the operator improved when (s)he was presented with depth information as opposed to only being presented with camera (2D) information. Depth information from Kinect sensor was overlaid on top of the live image from camera, reducing number of windows which test subject had to observe. Depth information was provided in a form of three transparent green arrows: one pointing directly forward and two at 25 degree angles to the left and to the right from central arrow (see Fig. 4). The length of the arrows was proportional to the distance to the nearest obstacle in direction of the particular arrow. Additionally, over each arrow a number representing distance in meters was added so to provide full information with minimal effort of the operator. In cases when obstacle distance was out of the range of sensing device, the arrow turned red and in that case last good measurement distance was displayed. Please note that this type of interface can easily be upgraded with additional arrows covering wider field of view since Kinect 1.0 has 57° horizontal field of view and angular resolution of 0.09°.

3) Operator tasks: The operators, regardless of used design setup, had to complete four tasks. In the first two tasks, the operators had to guide the robot to a desired position and orientation as accurately and as fast as possible (without emphasising either), both visually, without any feedback from the robot (T1), and in teleoperation mode (T2). The starting point, as well as desired position and orientation were the same in both tasks, and were marked on the floor. Both locations were known to test subjects. While teleoperating the robot, the test subjects could not see or hear the robot due to physical barriers (cardboard screen) and the fact they wore earmuffs. Time of completion, distance between desired and actual position (in direction of x and y axes) and orientation (angle) were measured, and number of collisions was counted (if any). It should be noted that there were no obstacles in the direct path of the robot (from start to goal position) but there were obstacles outside it, and some of the subjects collided

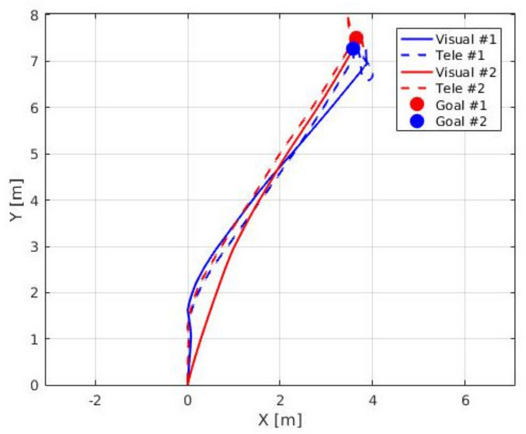


Fig. 5. Example of comparison of motion trajectories obtained using visual and teleoperaton conditions for two test subjects

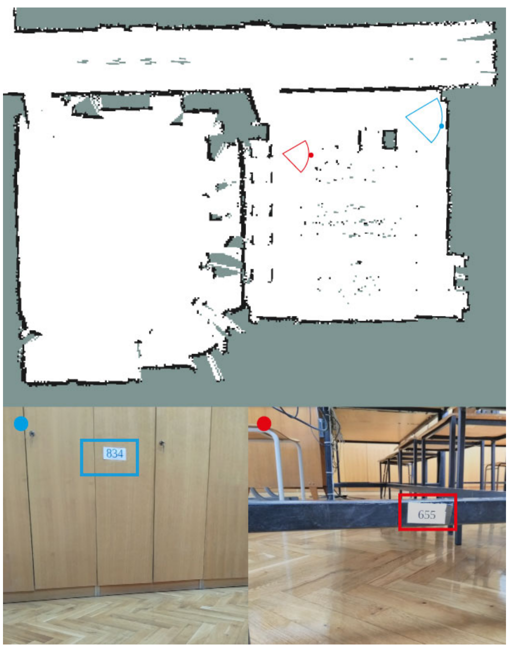


Fig. 6. Example of labels used for search and rescue task and their location on the map (marked in red and blue along with their locations on the map)

with them while trying to position the robot more accurately. Example of resulting robot motion trajectories can bee seen in Fig. 5 for two test subjects for tasks T1 and T2.

Another, more challenging, steering task in teleoperation mode (T3) was also performed. The goal of this task was the same as previous one, i.e. to steer the robot to given position and orientation as accurately and as fast as possible, but this time the route the operator had to take was significantly longer than in previous tasks (please see Fig. 2). Time of completion, distance between desired and actual position (in direction of x and y axes), and orientation (angle) were measured, as well as number of collisions.

The final task was a simulation of a simple search and rescue mission (T4) where the goal was to find as many objects as possible in a dislocated environment in a limited time frame of five minutes. In the experiment ten white labels (examples shown in Fig. 6) with unique random three-digit codes were printed in different font sizes (from 20 pt to 200 pt) and were attached to objects of interest that the test subjects had to find. Test subjects had to identify the codes. The labels were positioned so that they were concealed and that they were visible only from some (relatively) small part of the environment. This in turn resulted in the requirement that the operator had to steer the robot around the environment to a specific position and orientation to be able to identify it (just like trying to find victims in urban search and rescue operation). The time of completion of the task was measured (in case test subject successfully found all the objects before time limit), number of collisions, as well as number of identified labels (and which ones).

Tasks T1, T2, T3, and T4 were given to each test subject in predefined random order, so to compensate for possible learning effects.

B. Statistics

1) Two-Sample Student's t-test: This test was used to test the null hypothesis that means and variances of two population samples are equal.

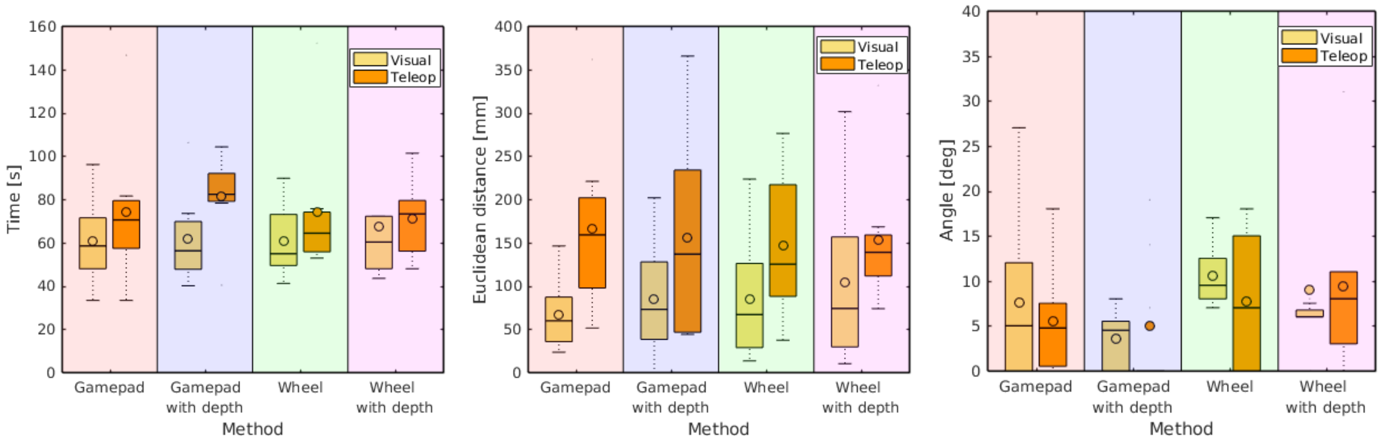
2) Kruskal-Wallis test: This test is a non-parametric test which aims to determine if medians of multiple groups are different (similar to ANOVA but without underlying assumptions). In case of our experiments, these groups are experiment setups. The null hypothesis for Kruskal-Wallis test is that population medians are equal. Rejecting the null hypothesis provides the information that there is significant difference between the populations, but will not indicate which ones are different and thus post-hoc testing needs to be performed (Tukey Honestly Significant Difference test in our case).

All statistics calculations were done at $\alpha = 0.05$ significance level. For more details about statistical tests, please refer to [16].

III. Results and Discussion

A. Comparison of visual and teleoperated steering

The analysis was performed on the results of steering tasks where test subjects had to steer the robot from initial to final position and orientation using both visual feedback and teleoperation. Summary of obtained results are presented in Fig. 7. They show that steering tasks were completed, on average, 20% slower in teleoperation, as well as that positioning error was almost doubled in teleoperation (per individual setup). However, both for T1 and T2, considering time of completion there is no statistically significant difference between the setups, according to Kruskal-Wallis test. This is a bit surprising, but might be due to relatively simple tasks which required short amount of time. Similar results were obtained for positioning error (Euclidean distance from the desired position to the measured position). The two-sample t-test was used to test if means of visual and teleoperation controls (for all three dependent variables) were different with



(a) Time of completion (b) Euclidean distance to desired position (c) Deviation from desired orientation

Fig. 7. Comparison of performance between visual feedback and teleoperation with different setups

statistical significance. Tests for all the setups, after applying Bonferroni correction for multiple comparisons, showed that samples do not differ, which might suggest that either tasks were relatively simple so differences did not emerged or that operators' situational awareness was good in both cases. However, when considering positioning errors along x and y axes separately, the two-sample t-test significant difference was observed only for steering wheel with depth test setup and only along y axis ($p = 0.0343$ after applying Bonferroni correction), but not along x axis. We believe this might be due to test setup configuration (i.e. distances between start and goal point in x and y direction), but further testing is required.

Considering number of collisions, a very low number of such events occurred, a total of 6 in T1 and 9 in T2. In T1 number of collisions was almost equal for each of the setups, while in T2 8 out of 9 collisions occurred in setups with depth, which might suggest that users were distracted with additional information which was presented to them.

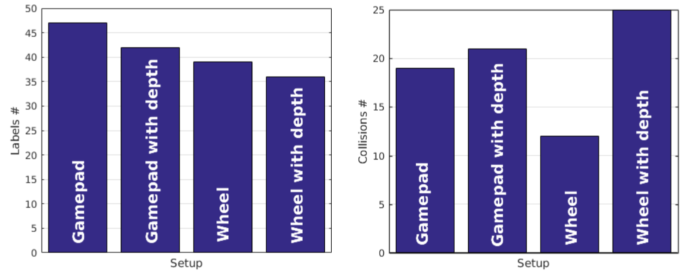
B. Assessment of teleoperation steering setup

In third task (teleoperation in more demanding conditions), there was also no statistically significant differences between the setups, in time of completion, deviation from desired final position and absolute deviation from desired final orientation. Results obtained in this task are presented in Table II, which demonstrates that, on average, subjects that used steering wheel completed their task faster than subjects using gamepad.

C. Search and rescue analysis

For search and rescue task, summarized results for all test setups are presented in Fig. 8. They demonstrate that for setup with steering wheel without depth information operators achieved less collisions (in total) than in other setups. However, no statistical significance was detected both for number of identified labels and for number of collisions (using Kruskal-Wallis test).

The maximum number of identified labels in this task was seven, while only one of the labels was not been found by any



(a) No. of identified labels (b) No. of collisions

Fig. 8. Search and rescue task results

of the test subjects. The average number of identified labels per setup was 5.875 for gamepad, 5.25 for gamepad with depth, 4.875 for steering wheel and 4.5 for steering wheel with depth. The average number of identified labels across all setups was 5.125.

D. Time delay

The time delay was measured for live camera feed and for depth feed. The summary of delay data, which was recorded with a resolution of up to 1 ns, is shown in Table III. From the table it can be seen than latency was on average similar across all test conditions.

IV. CONCLUSIONS

In the paper, the impact of user interface elements on teleoperation of mobile robot was studied with two control interfaces and two GUIs for a total of four setups. The experiment was designed as a between-subject user study in which a subject had to complete three steering tasks and a search and rescue task. Obtained results showed, as was expected, that all performance parameters were better for visual feedback control than for teleoperation mode (regardless of used setup). It was also observed that although various parameters degraded for teleoperation setups in most cases they did not have statistically significant difference, which was

TABLE II
RESULTS OF TELEOPERATION TASK IN T3

Setup	Time of compl.		Pos. error		Orient. error	
	μ $[s]$	σ $[s]$	μ $[mm]$	σ $[mm]$	μ $[°]$	σ $[°]$
Gamepad + GUI without depth	96.55	22.1	171.73	33.87	6.13	6.42
Gamepad + GUI with depth	106.50	22.27	138.43	75.38	7.37	7.00
Wheel + GUI without depth	92.10	30.02	138.65	81.48	4.25	4.53
Wheel + GUI with depth	92.94	16.32	172.56	101.11	10.50	9.13
TOTAL (mean)	97.02	22.8	155.34	75.1	7.06	7.01

TABLE III
TIME DELAY (IN SECONDS)

Setup	Live image		Depth	
	μ	σ	μ	σ
Gamepad + GUI without depth	0.0440	0.1700	0.0188	0.2049
Gamepad + GUI with depth	0.0432	0.1600	0.0179	0.1925
Wheel + GUI without depth	0.0445	0.1652	0.0186	0.1966
Wheel + GUI with depth	0.0455	0.2016	0.0209	0.2344
TOTAL (mean)	0.0443	0.1753	0.0191	0.2080

surprising. We believe that this lack of statistical significance might be in part due to the fact that sample size per condition was somewhat small and tasks were rather simple. Thus we plan to perform additional measurements in the future to obtain more general results. It is also interesting to note that addition of depth information did not improve the performance. In fact this might be due to the fact that users could (based on their previous experience and knowledge of relative object sizes) estimate depth information from live camera stream.

For the search and rescue task additional observations were made. While there were again no statistically significant results (with similar note as before) obtained result seem to suggest that gamepad interface resulted in more labels being found. This might suggest that robot was more dexterous with that particular control interface. Numbers of collisions were also more consistent across subjects for gamepad control interface than for steering wheel. However, steering wheel had lowest cumulative number of collisions (interestingly, for the case of no depth information, which might suggest that used depth information mode overloaded and distracted test subjects). While majority of our analyses showed no significant statistical difference between the setups, use of more advanced control devices and more complicated user tasks might produce statistically significant differences as was the case in [12].

In the future we plan to conduct similar research with more advanced HCIs (keyboard, VR, haptic feedback devices, etc.) and provide a map overlaid on top of live camera image, as was done in [11] instead of in separate window, to study the impact on teleoperation. Furthermore, we also plan to build a driver model based on control commands given by the operators, and study the impact of time delay on teleoperation of real mobile robots.

ACKNOWLEDGEMENT

The authors would like to express gratitude to all test subjects who participated in the user study.

REFERENCES

[1] A. Birk, K. Pathak, S. Schwertfeger, and W. Chonnaparamutt, “The iub rugbot: an intelligent, rugged mobile robot for search and rescue operations,” in *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR). IEEE Press*, vol. 10, 2006.

[2] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] V. Čelan, I. Stančić, and J. Musić, “Cleaning up smart citieslocalization of semi-autonomous floor scrubber,” in *International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. IEEE, 2016, pp. 1–6.

[4] J. M. Riley, D. B. Kaber, and J. V. Draper, “Situation awareness and attention allocation measures for quantifying telepresence experiences in teleoperation,” *Human Factors and Ergonomics in Manufacturing & Service Industries*, vol. 14, no. 1, pp. 51–67, 2004.

[5] M. R. Endsley, “Design and evaluation for situation awareness enhancement,” in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 32, no. 2. SAGE Publications, 1988, pp. 97–101.

[6] C. W. Nielsen and M. A. Goodrich, “Comparing the usefulness of video and map information in navigation tasks,” in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*. ACM, 2006, pp. 95–101.

[7] S. Smith-Atakan, *Human-computer interaction*. Cengage Learning EMEA, 2006.

[8] T. W. Fong, F. Conti, S. Grange, and C. Baur, “Novel interfaces for remote driving: gesture, haptic, and pda,” in *Intelligent Systems and Smart Manufacturing*. International Society for Optics and Photonics, 2001, pp. 300–311.

[9] L. A. Nguyen, M. Bualat, L. J. Edwards, L. Flueckiger, C. Neveu, K. Schwehr, M. D. Wagner, and E. Zbinden, “Virtual reality interfaces for visualization and control of remote vehicles,” *Autonomous Robots*, vol. 11, no. 1, pp. 59–68, 2001.

[10] S. Vozar and D. M. Tilbury, “Driver modeling for teleoperation with time delay,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3551–3556, 2014.

[11] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks, “Ecological interfaces for improving mobile robot teleoperation,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 927–941, 2007.

[12] J. Jankowski and A. Grabowski, “Usability evaluation of vr interface for mobile robot teleoperation,” *International Journal of Human-Computer Interaction*, vol. 31, no. 12, pp. 882–889, 2015.

[13] W. Li, Z. Liu, H. Gao, X. Zhang, and M. Tavakoli, “Stable kinematic teleoperation of wheeled mobile robots with slippage using time-domain passivity control,” *Mechatronics*, vol. 39, pp. 196–203, 2016.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[15] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[16] S. M. Ross, *Introductory Statistics*. Academic Press, 2010.