A Graphical User Interface for Multi-Robot Control in Urban Search and Rescue Applications

Farzad Niroui, Yugang Liu, *Student Member, IEEE*, Rafik Bichay, Elias Barker, Carlos Elchami, Owen Gillett, Maurizio Ficocelli, and Goldie Nejat, *Member, IEEE*

***Abstract*— This paper presents a human-robot interaction (HRI) interface designed to control heterogeneous multi-robot teams for urban search and rescue (USAR) applications. The interface allows an operator to configure and create teams from a list of available robots, and also monitor and control individual robots during the search mission. The motivation behind the design of this interface is to have a user-friendly interface which allows for situational awareness of all robots in a team with the hopes of minimizing the workload of the operator when controlling multiple robots. Experiments were conducted in a USAR-like environment with operators to investigate the efficiency of the interface using multiple heterogeneous robots. The results showed that the interface was effective in aiding operators to explore an unknown cluttered environment using all robots in the team while finding the majority of victims.**

Keywords—Graphical User Interace; Human-Multi-Robot Interaction; Urban Search and Rescue

I. Introduction

The use of robots can help rescue workers to speed up the time-critical emergency response operations in urban search and rescue (USAR) missions [1]. Compared to a single robot, multi-robot teams have the advantages of providing situational awareness from multiple locations, allowing for the implementation of low-cost distributed sensors, and improving the fault-tolerance of USAR missions [2]. While promising, the control of a team of rescue robots in cluttered and unstructured USAR scenes is still a challenging task as operators can easily get fatigued and lose situational awareness in such stressful operations [3]. To address this challenge, the development of user-friendly graphical user interfaces (GUIs) is important for the successful control of multi-robot USAR teams.

There has been previous work in designing GUIs for the purpose of control and information feedback of single rescue robots [4]-[11] and multiple rescue robot teams [12]-[14]. For example, in [4] a GUI customization was presented for a rescue robot being controlled using various modes in order to improve situational awareness, lower cognitive load and increase the efficiency of the operator. The GUI displayed a video feed of the front facing camera onboard the robot with panning, tilting and zooming capabilities as well as sonar and laser range information shown in colored boxes surrounding the camera feed to provide distance information. A generated 2D environment map could also be presented. A rear-looking camera placed on the robot provided rear video feed when needed above the front-facing camera feed. System alerts were also provided to indicate if sensor readings were not correct or battery level was low. Tests with five participants showed that the interface was easy to use.

In [5] and [6], a video-centric interface and a map-centric interface were compared. The video-centric interface consisted of two fixed video windows for front or rear facing cameras, a range panel to show sonar data, and a 2D map. The map-centric interface consisted of a 3D map with a robot avatar in the center of the interface. A 2D video was also displayed based on the pan-tilt position of the robot for the latter. Experiments with 8 participants comparing the two GUIs while controlling a robot were conducted in the National Institute of Standards and Technology (NIST) test arenas. Results showed that participants covered more area using the map-centric GUI. They preferred the location awareness of this GUI as it provided the use of both the 3D map with landmark marking and video. However, participants preferred the activity awareness of the video-centric GUI as it was easier to see the robot's movement capabilities using the 2D video. Participants also commented on what features they would keep. They preferred a 3D map over a 2D map, being able to track the progress of the robot by seeing the path it has taken, and having a fixed camera.

Based on the results obtained in [5] and [6], in [7], an updated GUI was designed. This GUI was a modification of the video-centric interface presented in [5] and [6]. The main difference was the addition of a new distance panel to replace the range panel. This panel used laser scans and sonar data to present the environment surrounding the robot in a top-down and perspective view. An experiment was conducted to compare the performance of this new GUI to the previous video-centric interface. Eighteen participants were asked to use both GUIs to traverse a robot through a narrow path and then use the same path to return. The completion time and the number of collisions were tracked and it was concluded that the new GUI outperformed the previous interface.

In [8], several GUIs used in robot rescue competitions using the NIST test arenas were discussed. Some observations of the human-robot interaction during the competitions assisted in providing general design guidelines such as using one monitor for the interface, the use of one robot to view another when multiple robots are available and designing for the intended user.

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), MacDonald, Dettwiler and Associates (MDA) Inc., and the Canada Research Chairs (CRC) Program.

F. Niroui, Y. Liu, R. Bichay, E. Barker, C. Elchami, O. Gillett and G. Nejat are with the Autonomous Systems and Biomechatronics Laboratory (ASBLab) in the Department of Mechanical and Industrial Engineering, University of Toronto, (email: farzad.niroui@mail.utoronto.ca), (e-mail: yugang.liu@mail.utoronto.ca), (e-mail: nejat@mie.utoronto.ca).

M. Ficocelli is with the Guidance, Navigation & Control group at MacDonald, Dettwiler and Associates (MDA) Inc., Brampton, Canada (e-mail: Maurizio.Ficocelli@mdacorporation.com).

In order to manage the use of multiple resources such as robots, dogs and soldiers for military village searches, a planning tool consisting of a resource allocation engine and a GUI was developed in [12]. Using existing shapefiles (for roads and buildings) of a search area, a user could create a village and determine a search plan including the search team, then display an animation of the plan via the GUI at a preferred playback rate. The GUI showed the map of the village and the user was able to adjust the search speed of the different search units for more insight on the search and could also modify the routine if necessary.

Two GUIs, one for a desktop PC and the other for a tablet PC, were presented in [13] in order to manage teams of both autonomous ground and aerial unmanned vehicles for the application of reconnaissance and surveillance. Using either GUI, action commands were first chosen by the operator, then the robots to execute the actions were selected, and lastly the operator could input geographic information for the action command. The GUIs displayed an aerial map of a surveillance area with the locations of the vehicles. Experiments were conducted in a field camp with four unmanned ground vehicles and two aerial vehicles. The two commands of move to a defined destination and observe a point of interest were tested. All 4 ground vehicles were able to autonomously move to a desired destination, and all vehicles were able to autonomously orbit around the point of interest and then position themselves in its proximity.

In [14] an interface was discussed for controlling a team of robots in the USARSim simulation environment. This GUI displayed the video feed of all the available robots in a list at the top of the window. The operator was able to click on the video feeds to control individual robots. In the middle of the interface, one panel showed an overall 2D map generated by the laser scanners onboard all the robots. A second panel showed a larger video feed of the selected robot, and a third panel displayed a robot specific 2D map. The operator gave commands to a robot by assigning waypoints on the robot specific 2D map, or by using a widget at the bottom of the interface to teleoperate the selected robot. Simulations were conducted with 15 participants to determine the effect of robot team size on human performance. Task performance increased from a team of 4 to 8 robots, but decreased with 12 robots. The increase of the number of robots resulted in workload increasing monotonically.

The majority of the aforementioned GUIs have been developed for single robot control with a few recent efforts focusing on GUIs for multi-robot team control in USAR applications. For the latter, how these interfaces can be designed for operators to generate robot teams has yet to be discussed. Furthermore, to the authors' knowledge the investigation of how the design of these GUIs effects situational awareness and the workload of an operator when controlling a team of multiple robots in a real environment is still lacking. In this paper, we present the development of a unique HRI interface for control of multi-robot USAR teams. The interface allows the operator to configure the available heterogeneous rescue robot resources at hand into search teams, and change these teams on the fly as needed. The operator can also monitor and switch between controlling each robot in the team of interest during a search mission. Preliminary experiments are presented herein to investigate the potential use of the interface.

II. HRI Interface Design for Multi-Robot Rescue Teams

The HRI Interface has been designed to allow for control of heterogeneous multi-robot teams. The robots in a team can effectively share information and complete tasks together, by utilizing the unique features of each robot. The overall system control architecture is presented in Fig. 1. The operator utilizes the interface to generate robot teams and also to directly obtain sensory information from the onboard robot sensors from the Robot Sensors module as well as maps of the USAR scenes from the SLAM (Simultaneous Localization and Mapping) module in order to monitor and control robot tasks. The level of autonomy of the robots can be changed from full teleoperation to semi-autonomous or autonomous control. The Autonomous/Semi-Autonomous module is used as the system decision making module for multi-robot team exploration and victim identification.

A. Design Requirements

The objective is to develop an interface that can: 1) provide the operator with a user-friendly graphical interface, 2) aid in maintaining situational awareness, and 3) minimize workload. The GUI should allow the operator to control the team, while being aware of the status of all the individual robots in the team. For the interface design, we used a number of the guidelines provided in [5], [6] and [8] including having only one monitor, having fixed camera views, promoting the remote view of one robot by another in the team, using both mapping and video information, and using an easy to use input device.

B. Layout

The layout of the interface consists of two windows: 1) the Robot Team Initialization Window, and 2) the Multi-Robot Control Window.

1) Robot Team Initialization Window

The Robot Team Initialization Window consists of a linear workflow, Fig. 2. The GUI initializes connection with all the available robots and prompts the operator to make teams with these robots. The operator adds the available robots to teams and confirms the teams. The team and individual robots are stored, and the Robot Team Initialization Window closes and the Multi-Robot Control Window opens. The layout for the Robot Team Initialization Window is shown in Fig. 3. The five main elements of the window are discussed in detail below.

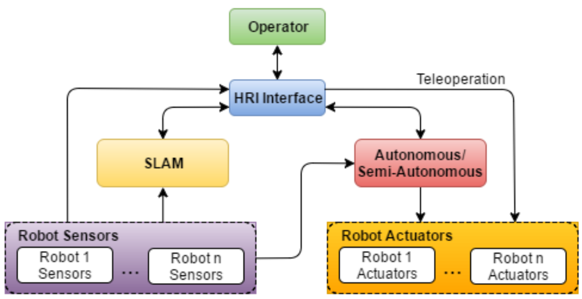

Fig. 1. Overall Control Architecture.

(1) *Available Robots/Team lists*: The Available Robots list displays all the available robots by their respective names. The Team list displays the current members of the team.

(2) *Find Connected Robots/Reset button*: When the Find Connected Robots button is clicked, a predefined list of IP addresses that are associated to different robots are pinged and the available robots are added to the Available Robots list. The Reset button is used to remove all actions taken so far with respect to the lists, and update the available robots list again.

(3) *Add/Remove buttons*: When the Add button is clicked, the highlighted robot will be moved from the Available Robots list to the Team list. Similarly, when the Remove button is clicked, the highlighted robot from the Team list will be moved back to the Available Robots list.

(4) *Confirm Team/Begin buttons*: When the Confirm Team button is clicked, the teams are generated and the Team list will be cleared. When the Begin button is clicked, the current window will be closed and the Multi-Robot Control Window will be opened.

(5) *Current Team drop-down list*: This list allows the operator to switch between teams in order to add or remove robots from a team.

2) Multi-Robot Control Window

The Multi-Robot Control Window is the main window of the interface. This window allows the operator to have access to all the functions available for interacting with the teams as well as individual robots within a team. Namely, the operator can switch between teams and individual robots. He/she can see the status of each robot and set the level of autonomy of the robots. The workflow for this window is presented in Fig. 4. The operator is able to select the team and the different robots in that team from a drop-down list. Selecting a robot will update the control window with the information for this particular robot. The operator can view the selected robot's status, which includes sensor and actuator information feedback in the form of video feed, environment map, and different customized widgets. The operator can also choose to manually control each robot in the team or he/she can command the search to initiate and the entire team will begin searching in autonomous/semi-autonomous mode. If the operator wants to end a search for a team while it is in progress, he/she can instruct the search to suspend. Lastly, the operator is able to halt all activities. The interface of the Multi-Robot Control Window contains eight elements, Fig. 5. These elements are discussed below.

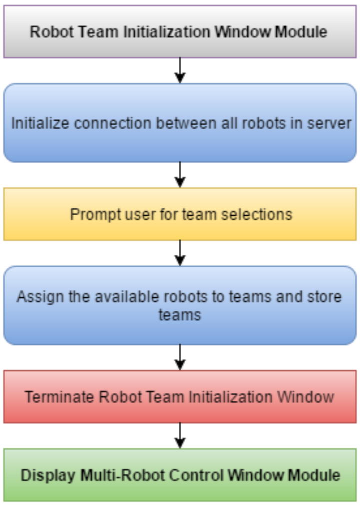

Fig. 2. Workflow of Robot Team Initialization Window.

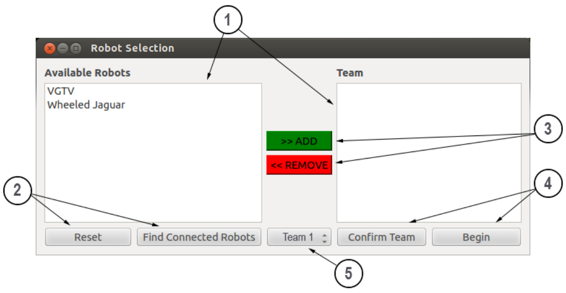

Fig. 3. Layout of Robot Team Initialization Window.

(1) *Overview Map of the Environment:* This area is used to display the generated map of the environment. Namely, either the option of displaying a 3D map from onboard 3D mapping sensors or a 2D map that is generated using laser scanners. The type of map is dependent on the onboard sensors of the individual heterogeneous robots. The map includes the location of all the team members and can be rotated and scaled.

(2) *Recent Message Logs:* The commands sent to the robots and the teams along with system alerts are displayed.

(3) *Robot Sensor and Actuation Display:* This widget shows the currently selected robot's velocity using sliders, and perimeter proximity information using boxes. For the latter, the white boxes turn red when the robot is too close to obstacles in the corresponding directions.

(4) *Current Robot Under Control Display:* Real-time video feed from an individual robot's onboard camera is displayed.

(5) *Status Indicators of Robots:* Indicates the status of individual robots. Green corresponds to a functioning robot, flashing red corresponds to a robot that is not functioning properly, and gray corresponds to an unavailable robot.

(6) *Team/Robot Selection drop-down lists:* This list allows the operator to select different teams and individual robots. Switching between robots will refresh the display to reflect the currently selected robot.

(7) *Autonomous/Manual Mode buttons:* Allows the operator to change the level of autonomy (teleoperation, semi-autonomous and autonomous modes) of a rescue robot.

(8) *Stop button:* When this button is clicked, all processes will be terminated.

C. Software

The interface was developed in Ubuntu using the Robot Operating System (ROS) and QT. A Microsoft Xbox 360 gamepad is used as the operator's input device to control the robots. The overall software architecture consists of 2 levels: the high-level for HRI and the low-level for robot control.

At the HRI level, the ROS package Joy was implemented to process the inputs from the human operator. We have developed the ROS package "Multi-Robot Control" which analyzes the messages obtained from Joy and determines:

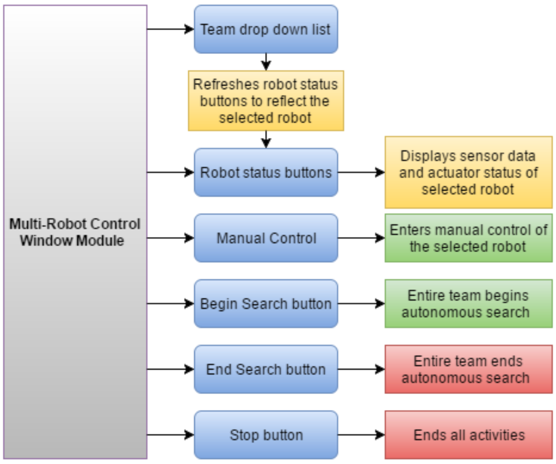

Fig. 4. Workflow of the Multi-Robot Control Window.

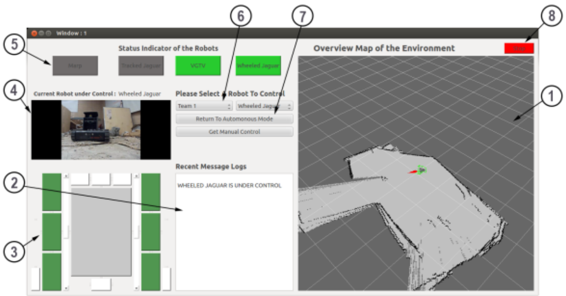

Fig. 5. Layout of Multi-Robot Control Window.

1) which robot is under control, 2) the selected control mode of the robot, and 3) the motion commands if the robot is under teleoperation. A ROS package "GUI Control" has also been developed to switch between the different widgets to show the status of each robot, the generated map, and corresponding video stream.

The low-level control consists of the specific robot control commands and mapping packages. A "Robot Control" package was developed for each robot to implement motion control commands. 2D video feed was obtained from the standard ROS package "openni_launch", while the 2D/3D mapping information was obtained using standard SLAM packages, such as "Hector_slam" for 2D and "RGBD_slam" for 3D.

III. Experiments

We conducted preliminary tests with the HRI interface in controlling a team of robots in a physical environment. The experiments were conducted in a cluttered USAR-like scene which was approximately 30m^2, Fig. 6. The objective was to have robot operators control a team of heterogeneous robots via the interface in order to explore the unknown environment and identify as many victims as possible.

The environment consisted of two separate sub-scenes containing items such as concrete, rocks, broken furniture, wood, and cardboard. Six dolls and mannequins, partially obstructed by rubble, were placed around the environment as victims. Five robot operators participated in the experiments. They did not have any previous knowledge of the environment layout or how many victims there were. During the experiments the sub-scenes were only visible to each operator through the GUI. The layout of the environment and the locations of the victims were the same for each operator.

A. Robots

Four robots were used in the experiments, Fig. 7: 1) the VGTV: a small tracked robot, 2) the MARP: a large six wheeled robot, 3) the Jaguar Lite: a large tracked robot, and 4) the Jaguar 4x4: a large four wheeled robot. The small VGTV robot was equipped with only a Sony FCB-IX11A 2D camera onboard and was tethered for both power and communication. Its polymorphic design and small size allows it to search small spaces. The MARP robot used an ASUS Xtion PRO Live camera for both 3D mapping and 2D video feed, and had a Hokuyo URG-04LX laser range finder for 2D mapping capabilities. Proximity sensors were also placed around its perimeter for obstacle detection. Both Jaguar robots used a Hokuyo URG-04LX laser range finder for 2D mapping and an ASUS Xtion PRO Live camera for 3D mapping and 2D video feed. The starting pose for each robot was consistent across all operator experiments. The VGTV robot was placed in the smaller sub-scene, and the MARP and Jaguar robots were placed in the larger sub-scene. We created this set-up to investigate if the operators would use the sensory information from the other robots in the same sub-scene to aid with exploration and if this would improve their situational awareness compared to the sub-scene with only the VGTV robot.

B. Experimental Procedure

Prior to the start of the experiments, each operator was provided with a training session in how to use both the interface and the gamepad. Then he/she was given 10 minutes to practice with the robots outside of the USAR-like environment. For each experiment, the operator was asked to launch the GUI, create a team using the four robots, and then conduct the scene exploration and victim identification tasks until he/she believed the tasks were completed.

Fig 6. The USAR experiment environment.

Fig 7. The four robots (From left to right: MARP, Jaguar Lite, Jaguar 4x4, VGTV).

The performance measures that were utilized for the experiments included: 1) percentage of scene explored, 2) number of victims found, and 3) overall mission time. To monitor the situational awareness of the operators, they were each asked to draw the map of the environment and locate where they believed the victims were found.

After the experiment, each participant completed the NASA Task Load Index (NASA-TLX) questionnaire [15]. The NASA-TLX consists of six subscales and the weighted average of these subscales can be used to estimate a user's workload for a specific task [15]. Each scale and the estimated workload are rated out of 100 with 0 being the best score and 100 being the worse score. The subscales are [15]: 1) *mental demand*: how mentally demanding was the task; 2) *physical demand*: how physically demanding was the task; 3) *temporal demand*: how rushed was the pace of the task; 4) *performance*: how successful was the operator in doing the task; 5) *effort*: how hard the operator had to work to accomplish his/her level of performance; and 6) *frustration*: how stressful and discouraging was the task.

C. *Results*

Each operator was able to use the interface to successfully create a team of four robots. Table I summarizes the performance results for each participant for the search task while using the robot team. A mock-up of the scene is provided in Fig. 8 and highlights the locations of the victims. In this figure the dashed rectangle indicates a ramp and the solid lines are used to show walls. The victim numbers in Table I correspond to the same numbers used in Fig. 8. The average number of victims found was 4.6 and the average percentage of scene explored was 67.4%. Each victim was found at a different time and in a different order depending on the search strategy the operators used.

Table I: User victim identification and overall mission time in seconds and percentage of scene explored (- indicates a victim was not found).

Victims	Participants				
	1	2	3	4	5
1	1000	1280	180	510	300
2	28	540	-	-	-
3	434	840	551	255	538
4	590	405	778	450	110
5	1630	-	-	-	-
6	170	40	314	598	740
End Time	1861	1825	1365	732	851
% Explored	84%	68%	81%	51%	53%

Fig. 9 shows the maps that each participant had drawn (in red) overlaid on top of the mock-up of the environment. The blue areas indicate the locations that were explored with robot paths shown in dark blue. We use these maps to investigate the situational awareness of the operators. As can be seen in the figure, participants 1, 3, 4 and 5 were able to approximately identify the locations of the victims they found. Furthermore, these four participants were also able to estimate the locations of barriers and slopes they encountered within the scene such as the walls and ramps. Participant 2, even though asked to do so, did not draw the scene and only located the victims. When using the Jaguar 4x4, Jaguar Lite, and MARP robots, the operators had better situational awareness. This is due to the fact that they had access to three different camera views and maps of the same sub-scene, which allowed them to see the other robots and use this to aid in their search. As expected, since the VGTV robot only had a 2D camera view, the operators did not have the same situational awareness of the sub-scene it was in and pre-maturely believed they had explored the surrounding area.

Each participant used all four robots to explore the environments. The minimum number of switches between the robots was determined to be four. The two Jaguar robots were used the majority of the search.

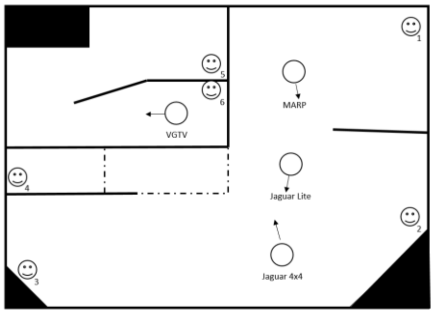

Fig. 8. Mock-up of the scene with the locations of the victims and the initial locations of the robots.

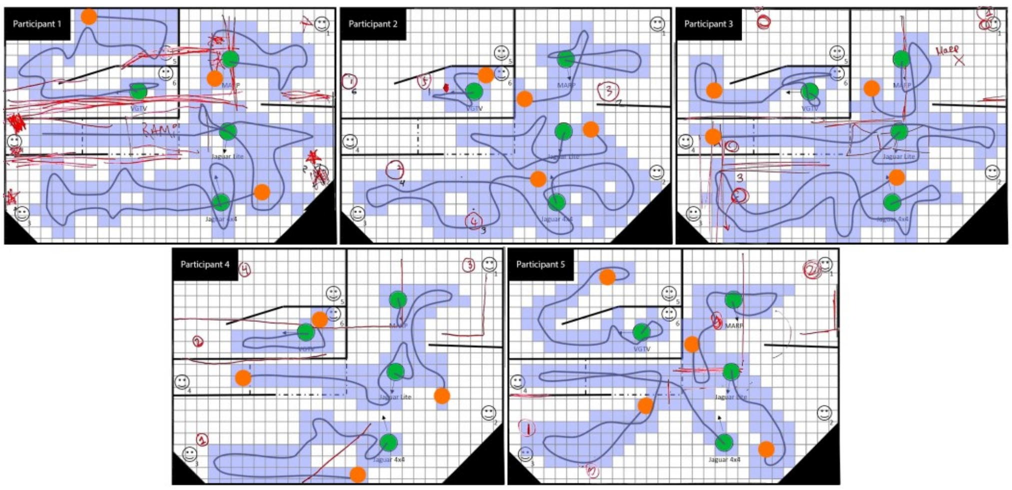

Fig. 9. Map drawn by each participant (in red), the area they explored (light blue), path taken (dark blue), and start (green) and end (orange) locations.

The results show that the interface did enable the operators to conduct a search using all four robots. Participants 1 and 2 were the most successful in finding the victims and also had the longest search times. Exploring more of the environment did not necessarily result in better performance, but rather a well-planned search strategy helped, as can be seen when comparing the results of Participants 2 and 3.

1) Questionnaire Results

The results of the NASA TLX questionnaire are presented in Fig. 10. With consideration of designing a user-friendly interface that can help maintain situational awareness, the subscales of mental demand, performance, effort, and frustration level were given an equal weighting of 0.2. As temporal demand refers to the time pressure felt due to the pace of the task, it has a weighting of 0.1. As previously mentioned, the participants did not have a time limit. Similarly, physical demand also had a lower weighting of 0.1.

The final workload results are presented in Table II based on the aforementioned weightings. The average workload was determined to be 58.5, which is defined to be moderate. This workload level is understandable as for these experiments, the operators were teleoperating all four robots. We postulate that by providing more autonomy to the robots this will lower the workload level. This is a part of our current work [2].

IV. Conclusion

In this paper, the design of an HRI interface for controlling multi-robot teams of heterogeneous rescue robots was presented. An operator is able to configure and create these robot teams from a list of available robots and also monitor and control individual robots during the search mission. The motivation behind the design of this GUI was to have a user-friendly interface, which allowed for situational awareness of the robots in a team and minimized the workload of the operator. Experiments conducted with the interface and a team of four heterogeneous robots in a USAR-like environment are promising. Future work will include fully incorporating the robot semi-autonomous/autonomous module in order to increase the autonomy of the robots in the team, and then conducting more vigorous tests of the functionality of the interface for multiple teams and larger teams of robots, as well as a comparison of different GUI designs.

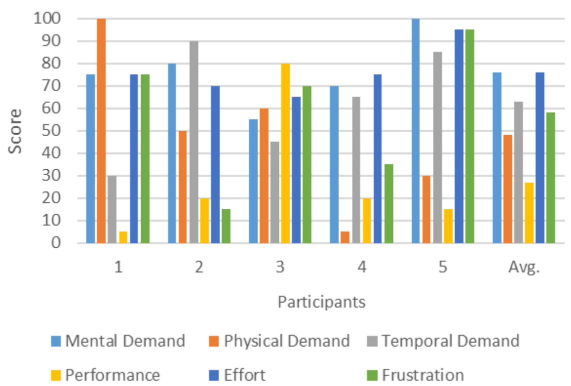

Fig. 10. NASA TLX questionnaire results.

Table II. Weighted subscale values and estimated workload from NASA TLX Questionnaire.

Subscales:	Participants:						
	Weight	**1**	**2**	**3**	**4**	**5**	**Avg.**
Mental Demand	0.2	15	16	11	14	20	15.2
Physical Demand	0.1	10	5	6	0.5	3	4.8
Temporal Demand	0.1	3	9	4.5	6.5	8.5	6.3
Performance	0.2	1	4	16	4	3	5.4
Effort	0.2	15	14	13	15	19	15.2
Frustration Level	0.2	15	3	14	7	19	11.6
Workload:		59	51	64.5	47	72.5	58.5

References

[1] Y. Liu and G. Nejat, “Robotic urban search and rescue: a survey from the control perspective”, *Journal of Intelligent and Robotic Systems*, vol. 72, no. 2, pp. 147-165, 2013.

[2] Y. Liu and G. Nejat, “Multi-robot cooperative learning for semi-autonomous control in urban search and rescue applications”, *Journal of Field Robotics*, vol. 33, no. 4, pp. 512-536, 2016.

[3] J. Casper and R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center,” *IEEE Trans. Syst. Man, Cybern. B, Cybern*, vol. 33, pp. 367-385, 2003.

[4] M. Baker, R. Casey, B. Keyes, and H. A. Yanco, “Improved interfaces for human-robot interaction in urban search and rescue,” in *Proceedings of the IEEE Conf. on Syst., Man, and Cybern.*, 2004, pp. 2960–2965.

[5] J. L. Drury, B. Keyes, and H. A. Yanco, “Lassoing hri: Analyzing situation awareness in map-centric and video-centric interfaces,” in *Proceedings of the ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2007, pp. 279–286.

[6] H. A. Yanco, M. Baker, R. Casey, B. Keyes, P. Thoren, J. L. Drury, D. Few, C. Nielsen and D. Bruemmer, “Analysis of human-robot interaction for urban search and rescue,” in *Proceedings of the IEEE Int.Workshop on Safety, Security and Rescue Robotics,* 2006.

[7] H. A. Yanco, B. Keyes, J. L. Drury, C. W. Nielsen, D. A. Few, D. J. Bruemmer, "Evolving Interface Design for Robot Search Tasks," *Journal of Field Robotics,* vol. 24, pp. 779–799, 2007.

[8] H. A. Yanco, J. L. Drury, "Rescuing interfaces: a multi-year study of human-robot interaction at the AAAI robot rescue competition," *Journal of Autonomous Robots,* vol. 22, pp. 333–352, 2007.

[9] B. Larochelle, G. J. M. Kruijff, N. Smets, T. Mioch, and P. Groenewegen, “Establishing human situation awareness using a multi-modal operator control unit in an urban search rescue human-robot team,” in *Proceedings of the IEEE Int. Symp. on Robot and Human Interactive Communication*, 2011, pp. 229–234.

[10] H. Martins and R. Ventura, “Immersive 3-d teleoperation of a search and rescue robot using a head-mounted display,” in *Proceedings of the IEEE Conf. on Emerging Technologies Factory Automation,* 2009, pp. 1–8.

[11] A. Sandygulova, A. G. Campbell, M. Dragone, and G. O’Hare, “Immersive human-robot interaction,” in *Proceedings of the ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2012, pp. 227–228.

[12] R. Friese, P. Maxwell, A. A. Maciejewski, and H. J. Siegel, “A graphical user interface for simulating robust military village searches,” in *Proceedings of the Int. Conf. on Modeling, Simulation and Visualization Methods,* 2011, pp. 75–81.

[13] M. Langerwisch, T. Wittmann, S. Thamke, T. Remmersmann, A. Tiderko, and B. Wagner, “Heterogeneous teams of unmanned ground and aerial robots for reconnaissance and surveillance - a field experiment,” in *Proceedings of the IEEE Int. Symp. on Safety, Security, and Rescue Robotics,* 2013, pp. 1–6.

[14] P. Velagapudi, P. Scerri, K. Sycara, H. Wang, M. Lewis, and J. Wang, "Scaling effects in multi-robot control," in *Proceedings of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 2121–2126.

[15] S. G. Hart, “NASA-Task Load Index (NASA-TLX); 20 Years Later,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 2006, pp. 904–908.