Multi-modal User Interface for Multi-robot Control in Underground Environments*

Shengkang Chen,[1] Matthew J. O'Brien,[2] Fletcher Talbot,[3] Jason Williams,[3]
Brendan Tidd,[3,4] Alex Pitt[3] and Ronald C. Arkin[2]

Abstract— **Leveraging both the autonomy of robots and the expert knowledge of humans can enable a multi-robot system to complete missions in challenging environments with a high degree of adaptivity and robustness. This paper proposes a multi-modal task-based graphical user interface for controlling a heterogeneous multi-robot team. The core of the interface is an integrated multi-robot task allocation system to allow the user to encode his/her intents to guide the heterogeneous multi-robot team. The design of the interface aims to provide the human operator continuous situational awareness and effective control for rapid decision-making in time-critical missions. Team CSIRO Data61 came in second place utilizing this interface for the DARPA Subterranean (SubT) Challenge. The ideas used for this user interface can apply to other multi-robot applications.**

I. Introduction

Heterogeneous multi-robot systems have drawn more attention in recent years because of the unique capabilities in a wide range of applications including exploration, search and rescue, and hazardous area inspection, especially for environments that are too dangerous or inaccessible to humans. DARPA's Subterranean Challenge (SubT) [1] was a multi-year international challenge that aimed to encourage the development of multi-robot systems deployed in unknown underground environments. In SubT, a team of robots, supervised by a single human operator at a remote work station, needed to navigate and locate artifacts (e.g., helmets, tools and backpacks) in an unknown and unstructured underground environment. These environments are challenging for robots because of the unreliable communication, lack of GPS for localization and difficult terrain. The goal of SubT was to promote research on robots in an underground environment for time-critical missions (e.g., search and rescue). Team CSIRO Data61, composed of Emesent, Georgia Tech and CSIRO, deployed a heterogeneous multi-robot team including tracked robots, legged robots, and UAVs for the challenge. Due to the nature of underground environments, robots with a high level of autonomy can operate with limited or even no human intervention, providing unique capabilities. The overall approach of Team CSIRO Data61 in phase 1 (the Tunnel Circuit in August 2019) and phase 2 (the Urban Circuit in February 2020) is detailed in [2].

Compared with fully autonomous systems, semi-autonomous systems can take advantage of human expert knowledge and contextual decision-making to achieve better robustness and adaptivity. An unintuitive and poorly-designed user interface could make the operator misinterpret the situation and provide incorrect instructions to robots, which would lead to mission failures. As a result, it is critical to develop a user interface that can provide continuous situational awareness and effective control of multi-robot systems. To achieve this goal, we have developed a multi-modal user interface for controlling a heterogeneous multi-robot system. The interface provides various levels of control options from task assignment between robots to teleoperation. The operator can switch to different control options seamlessly if needed. Task-based control is the default mode of control. In task-based control, robots are coordinated using multi-robot task allocation (MRTA), where robots can acquire available tasks based on their heterogeneous capabilities and execute them automatically. Moreover, it avoids continuous control from the operator, which reduces operator workload and allows better decision-making. As a result, we developed the user interface specifically for task-based control. In task-based control, the interface displays tasks as pin markers and robot models in a 3D map. The operator can interact with these makers directly for intuitive control. Combined with audio messages about robot states, the interface can provide a clear overview of the current state of the environment and robots continuously.

This multi-modal graphical user interface has been used and tested in complex real-world environments. Team CSIRO Data61 used this interface for SubT in multiple underground environments including caves, tunnels and urban underground. Team CSIRO Data61 achieved second place in the final event and demonstrated the effectiveness of the user interface. This interface can also be applied in various multi-robot applications including exploration and reconnaissance.

*This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

[1]Shengkang Chen is with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA schen754@gatech.edu

[2]Matthew J O'Brien and Ronald C. Arkin are with the School of Interactive Computing, Georgia Institute of Technology, Atlanta, Georgia, USA mjobrien@gatech.edu, arkin@gatech.edu

[3]Fletcher Talbot, Jason Williams, Brendan Tidd and Alex Pitt are with the Robotics and Autonomous Systems Group, CSIRO, Pullenvale, Queensland, Australia fletcher.talbot@csiro.au, jason.williams@csiro.au, brendan.tidd@csiro.au, alex.pitt@csiro.au

[4]Brendan Tidd was with School of Electrical Engineering and Robotics, Queensland University of Technology, Brisbane, Queensland, Australia when this work was completed.

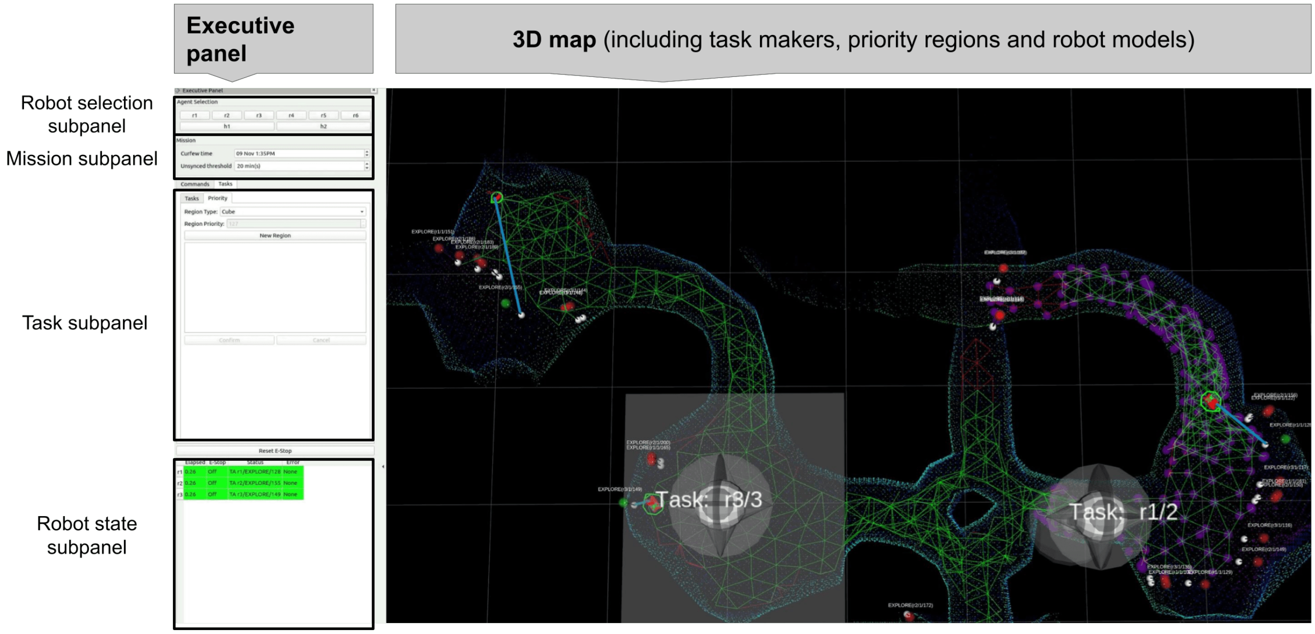


Fig. 1: An overview of the task-based user interface, which includes an executive panel on the left and a map-based interface on the right. The map is generated using local navigation data and SLAM data. The map shown is composed of a point cloud map and a traversability map. The operator can use the executive panel to generate tasks and priority-regions, which are shown on the map as pin markers and large gray shapes with markers on top respectively. Green colored markers are the executing tasks and red colored ones are the unassigned tasks. Robots are highlighted by the green hexagons and connected to their executing tasks (green colored markers). The area in the traversability map highlighted by purple spheres are the prioritized by graph-based priority-region (the large gray sphere, labeled Task:r1/2).

II. Background

Multi-robot task allocation (MRTA) is a critical part of multi-robot coordination to improve the overall system performance. In multi-robot task allocation, each robot is assigned to a subset of tasks for efficient task execution. For SubT, Team CSIRO Data61 developed a decentralized task allocation framework for multi-robot coordination using consensus-based decentralized auctions [3]. Since it is difficult for multi-robot teams to operate fully autonomously in real-world environments, incorporating human knowledge and supervision can improve the robustness and performance of the overall system [4], [5]. The task allocation framework offers specific ways to inject human knowledge, which will be covered in the following section.

Human multi-robot interaction and interfaces are the key part of human supervised multi-robot teams. However, controlling multiple robots simultaneously is challenging for the human operator due to the high workload [6] and context switching [7]. Researchers have developed various graphical user interfaces for different scenarios from hazardous environment intervention [8] to search and rescue [9]. In addition to conventional graphical user interfaces, various alternative interaction approaches have been proposed including gesture-based control [10], dialogue-based control [11], multi-figure touch [12] and virtual reality (VR) [13], [14]. VR can be an alternative to conventional graphical user interfaces since it helps increase users' situational awareness. However, context switching between different robots with various levels of control can cause disorientation, which makes it unsuitable for this application.

III. Interface Design

As mentioned above, SubT requires each team to deploy a team of robots supervised by a single human operator to locate as many artifacts (e.g., survivors, helmets, tools and backpacks) as possible within a time limit to mimic time-critical missions. During missions, the operator needs to make critical decisions in a timely manner. It is crucial for the user interface to provide both continuous situational awareness and effective control of a heterogeneous multi-robot team. As a result, we design the user interface to be able to provide a clear overview of the current progress including robot state, available tasks and the map of the environment, and allow the operator to inject human knowledge to guide the multi-robot team for successful mission completion.

To avoid continuous low-level control of each robot (which is impractical in a communication-constrained environment and inefficient for multi-robot systems), robots need to have a high-level of autonomy for coordination. Based on the HRI level framework from [15], our graphical user interface (Fig. 1) provides options from supervisor level for multi-robot coordination control to mechanical level for teleoperation as a backup option for a higher degree of robustness. There are two modes of interaction at the supervisor level: task mode and command mode. Task mode as the default mode of operation formulates multi-robot coordination as a multi-robot task allocation (MRTA) problem, where the mission is automatically decomposed to tasks for robots to complete, and the operator (optionally) manages robots by influencing the task allocation process. The command mode allows the operator to create and send a list of commands to a selected robot to follow. The overall system is ROS-based; RVIZ and QT libraries are used to implement the interface.

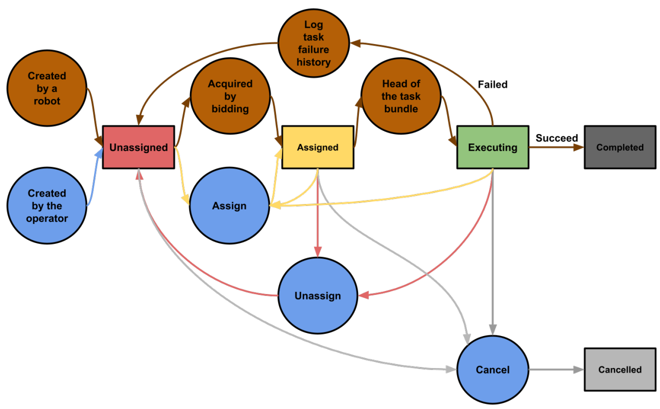


Fig. 2: The state transition diagram of a task. The task can be controlled by the autonomous robot actions (brown circles) and manual operator override actions (blue circles). For the assign action, the operator can assign the task to a robot even it is already assigned to another robot or it is executing.

The heterogeneous multi-robot team of CSIRO Data61 includes three types of robots: tracked robots (BIA5 Ozbot All-Terrain Robots or ATRs) which can carry and deploy communication nodes to build a mesh communication network; legged robots (Boston Dynamic Spots) which can navigate through challenging terrain and narrow cave corridors; and drones (Emesent Hovermaps) which can search areas inaccessible for ground robots.

In task allocation, a decentralized market-based approach was implemented. Based on the Consensus-Based Bundle Algorithm (CBBA) [3], robots use a consensus procedure to perform decentralized auctions of tasks and maintain a consistent task set. For unassigned tasks, robots submit bids based on their task execution cost, time-discounted reward and failure history. Bids from any robot are broadcast to connected agents, and each robot performs the same auction logic internally to determine the winner. As long as robots are communicating, they will form a consensus on who owns which tasks. This enables the task allocation to continue even when a subset of robots is disconnected from the communication network. While these disconnected groups may create conflicting assignments (e.g., two different agents own the same task), when communication is re-established the consensus rules will resolve the conflict (e.g., the agent with the lower bid drops the task). During the task allocation process, robots continuously bid on tasks and add them to their task bundle, which is an ordered list of tasks to complete. These tasks are generated by both the robots autonomously and the operator. Fig. 2 shows how robots change the state of a task to achieve effective coordination. Four types of tasks were used in SubT, where two of them can only be created by the operator:

- Explore task: perform frontier exploration.
- Sync task: perform data sync between robots and the base station.
- Drop node task (manually generated): drop a communication node for expanded communication coverage.
- Goto task (manually generated): drive to an assigned location.

The bid on a task is comprised of both a reward that represents the utility of the given robot executing the task, and a priority level. A higher priority is equivalent to an infinitely higher reward, and higher priority tasks will always be executed first. Priority levels offer a useful lever for operators, as estimating accurate rewards can be laborious even offline. Rather than adjusting rewards on the fly, the operator can change the priority level to ensure that certain tasks are completed first or last.

The operator's base station runs a modified version of the task allocation process using the same consensus protocol to maintain a consistent task set with the robots in the team, and forward bids between agents. The task allocator in the base station can relay task information including task status, task location and task type. Moreover, the operator can interact with the task allocators of robots via the task allocator in the base station to update the task set (create new tasks, change the status of existing tasks, and adjust priorities of tasks), and these changes are then broadcast to the robots. However, since the base station cannot execute tasks like robots, its task allocator will not bid on the tasks itself.

Fig. 1 shows an overview of the graphical user interface that includes a map-based interface and the executive panel. The map-based interface provides an intuitive visualization of tasks and robots within a representation of the world, allowing the operator to see their location and other state information at a glance. The map is generated using SLAM and navigation data in real-time during missions. The executive panel contains four main parts: the robot selection subpanel, mission subpanel, task subpanel, and a robot state table. Since task mode is the default mode of operation, the human operator primarily uses the task subpanel. The robot selection subpanel is used to select the corresponding robot(s) for the action the human operator wants to perform. The mission subpanel is for setting the time threshold for operating out of communications, and a final mission curfew time, to control when robots will generate a sync task and bring data back to the base station. This feature enables the operator to balance between deep exploration and updating the base station, as the mission requires. The task subpanel is for creating tasks and prioritization regions. Prioritization regions can change the priority of multiple tasks. The robot state table is to show the current state of robots.

With tasks shown in a 3D map as pin markers, the operator can control task allocation at both a single-task level interacting with task markers in the 3D map and a region (multi-task) level using prioritization regions to modify the priorities of included tasks. The flow of control on a task is shown in Fig. 3. For creating a new task, the operator selects the task type from the task subpanel and its location in the 3D map. For controlling an existing task, the operator needs to select an operation in the context menu of the corresponding task marker. The control flow at the region (multi-task) level has a similar structure without setting the task type, instead, the operator needs to set the region type and priority.

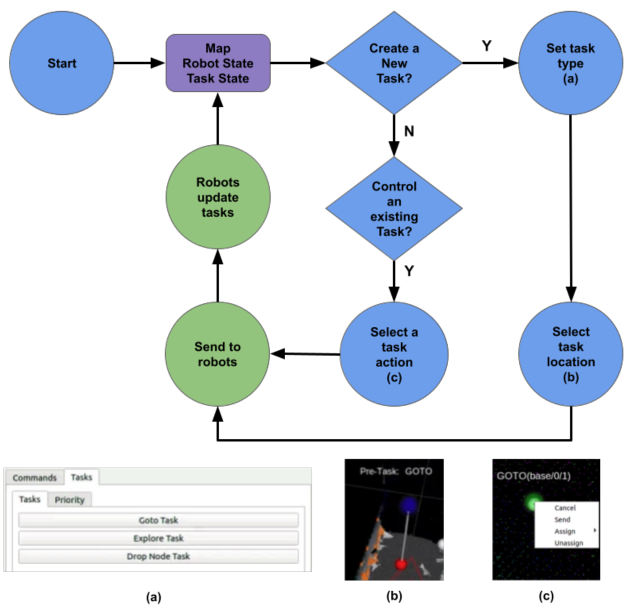


Fig. 3: The task-based control flow at a single-task level. The blue nodes are the actions and decisions made by the operator, while the green nodes are the actions executed by the base station and the robots automatically. Each operator action is shown with the corresponding part of the GUI. To create a new task, the operator first selects the task type from the task subpanel (a). Then, the operator will choose the task location in the 3D map using the pre-task marker. The pre-task marker (b) previews the task location and the task type before sending the task to the robots. To act on a task, the operator will use the context menu of the corresponding task marker (c). The control flow at a region (multi-task) level has a similar structure without setting the task type.

A. *Situation Awareness Enhancement*

One of the main goals of this multi-modal graphical user interface is to provide the operator with continuous situation awareness about the mission progress and the environment. Compared to a camera-based user interface, a map-based user interface requires a lower operator workload on multi-robot navigation tasks [16]. Furthermore, since the use of models and markers has been shown to be effective for providing the operator perception and comprehension of important information of the environment [17], [18], the interface adapts a map-based approach, where the 3D map of the environment shows locations and state information through task markers and robot models. The 3D map is generated and updated continuously using SLAM data and navigation data from robots. Each task marker is an interactive marker and represents a task, where the color of the marker indicates the state of the task and the text above the marker shows the name of the task. The name of the task contains the information about task type, task creator, and task ID. For instance, $EXPLORE(r1/1/10)$ indicates this is an exploration task represented by task type equals to 1, and it is created by the robot $r1$ with the task ID of 10. Left-clicking a task marker shows additional information including its assigned robot, its bid value and its priority as shown in Fig. 5a. Blue line segments are used to connect tasks in a robot's task bundle to indicate their execution order (Fig 1). In the case of only one task in the robot's task bundle, there will be a single blue line segment connecting the robot to its executing task. In the case of multiple tasks in the robot's task bundle, there will be a path containing multiple blue line segments to connect the robot model and its bundle's tasks (executing task and assigned tasks) in execution order. This map-based interface allows the operator to have a clear overview of the multi-robot team's current progress in a complex and partially unknown environment.

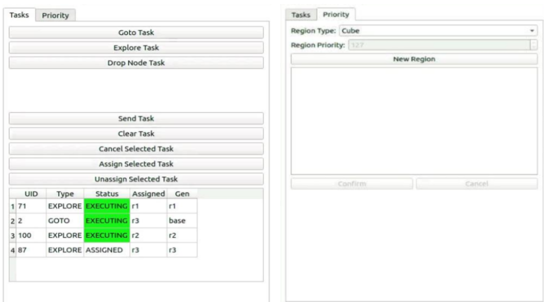
The robot state table in the executive panel shows the current state of each robot through four columns: elapsed indicates time passed since the last message received by the base station, E-stop indicates if the emergency stop is activated; status represents the current action the robot is performing; and error contains additional information if the robot encounters errors. Cells in each table will be filled with various colors to reflect the state, where green signifies a normal state, yellow signifies a warning state, and red signifies an error state. As a result, the operator can have a good understanding of the robot state at a glance.

Since the multi-modal user interface can improve human operator concurrent task processing efficiency by leveraging multi-channel communication [19], the multi-modal interface will send an audio message when the robot is in a warning or error state. This reminds the operator to switch back to supervise the multi-robot team from artifact identification. Since the audio message contains information about which robot is in a warning or error state, it also helps the operator to examine the robot needing attention when focusing on a different robot.

These features aid the operator's understanding about the current state of tasks and robots in the environment intuitively without the need of context switching between robots as recommended by the guidelines in [18] on creating an at-a-glance display and using multi-modal alert techniques to maintain situational awareness.

B. *Injecting Operator Knowledge*

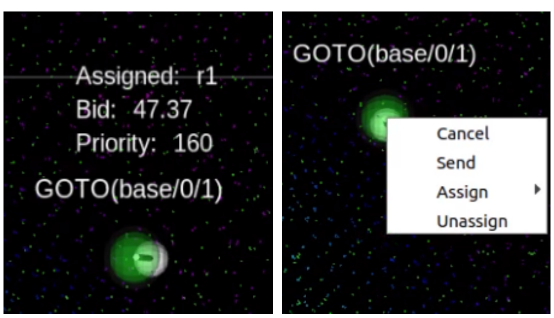
Along with providing good situation awareness, another goal of the graphical user interface is to allow the operator to inject his/her knowledge conveniently during the mission for effective control. To do so, the operator can use the interface to manipulate tasks at both the single-task level and the region level to influence task allocation and guide the multi-robot team.



(a) Task tab (b) Priority tab

Fig. 4: The task subpanel of the executive panel contains a task tab for create tasks and a priority tab for creating the prioritization regions

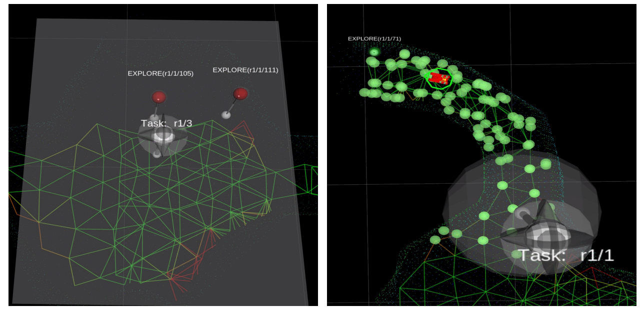
Robots automatically generate tasks during a mission, and the operator can also generate tasks using the task tab (Fig. 4a). These manually created tasks have a higher priority by default. To control single tasks, other than left-clicking it for additional information (Fig. 5a), the operator can create/cancel/assign/unassign a task directly using the context menu of the corresponding interactive task marker by right-clicking it (Fig. 5b). These actions will change the state of a task during the task allocation process as shown in Fig. 2. Before the introduction of interactive markers, an operator needed to rely on the task table in the executive panel to find the corresponding task from the map and control it using the buttons in the task tab. This improvement makes single-task level control more intuitive and streamlines the control process.



(a) Left-clicking (b) Right-clicking

Fig. 5: To interact with the task marker, left-clicking shows additional information while right-clicking allows task-level operations.

The operator can create prioritization regions using the priority tab of the task subpanel (Fig. 4b) to change the priorities of multiple tasks. These regions can be specified for a specific type of task or a specific subset of robots using the agent selection subpanel. This method can attract robots to execute tasks in an area the operator believes is important by setting a high-prioritization region, or repel robots from executing tasks in an unimportant or dangerous area using a low-prioritization region. There are two types of prioritization regions. One of them is a geometric prioritization region (Fig. 6a), represented either as a cube or half-plane to specify an exact volume or infinite boundary respectively. These modify the priority of tasks within the region. The other type is a graph-based prioritization region (Fig. 6b) represented as a sphere region. It modifies the priorities of tasks that are inside or downstream of the prioritization region in the topometric graph commencing from the origin (the location of the base station in SubT). The topometric graph is a coarse representation of traversability through the world used for planning, and is generated by combining local navigation data and SLAM data sequentially. Fig. 6b highlights the vertices of the topometric graph impacted by the prioritization region, where tasks located near these vertices will be prioritized.



(a) Geometric (b) Graph-based

Fig. 6: The two types of prioritization regions: geometric prioritization regions and graph-based prioritization regions. Geometric prioritization regions modify the priorities of tasks within its geometric shape, whereas graph-based prioritization regions modify the priorities of tasks near the highlighted topometric graph vertices.

These two types of prioritization regions are suitable for different scenarios. Geometric prioritization regions are more suitable for known environments. The operator can create a geometric prioritization region over an important/dangerous area so that robots will prioritize/deprioritize tasks within the region. Alternatively, graph-based prioritization regions are more suitable for partially unknown environments. Robots prioritize both the tasks within the graph-based prioritization region and tasks downstream of the region deeper into the environment. This is particularly useful for exploration in complex environments like tunnels. A graph-based prioritization will continue to guide robots deep into the unknown environment for exploration.

IV. Results

The user interface has been tested in various challenging environments and demonstrated its advantages in maintaining situational awareness and high-level control to aid the human operator for rapid decision-making. Team CSIRO Data61 achieved second place in the Systems Track using this interface.

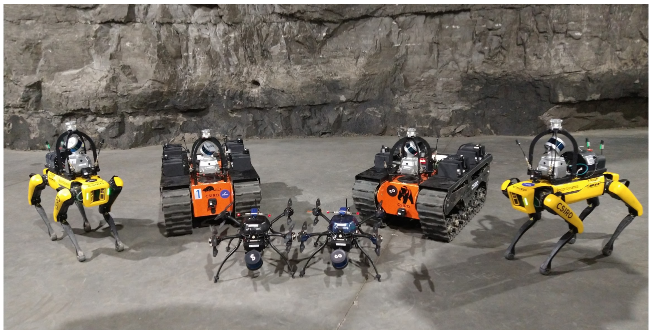


Fig. 7: The heterogeneous multi-robot team of Team CSIRO Data61 for the DARPA Subterranean Challenge Final Event: two BIA5 OzBot ATRs (tracked robots), two Boston Dynamics Spots (legged robots) and two Emesent Hovermap drones.

Robot	Autonomous	Guided Control	Direct Control
Rat-r1 (ATR)	65.84%	4.13%	30.03%
Bear-r3 (ATR)	12.46%	15.53%	72.01%
Bluey-r2 (Spot)	14.11%	72.76%	13.13%
Bingo-r5(Spot)	10.50%	72.94%	16.56%

TABLE I: The distributions of various levels of controls (autonomous, guided control, direct control) in runtime for each ground robot during the final runs. At the autonomous level, the robot was executing default tasks autonomously without any human interventions. A robot under guide control indicates it was executing tasks guided by the operator via task-based control (manual tasks and prioritized tasks). A robot under direct control indicates it was executing commands from the operator or being teleoperated.

Fig. 8: The activities of each ground robot during the final run. Prioritized tasks were the tasks modified by prioritization regions; manual tasks were the tasks manually assigned to the robot by the operator, default tasks were the tasks with unchanged priorities. Note that Rat failed to operate after 40 minutes and Bingo failed to operate after 20 minutes due to inability to recover from falls.

A. DARPA Subterranean Challenge Finals

The DARPA Subterranean Challenge Final Event was hosted in the Louisville Mega Cavern. The underground environments include tunnels, urban settings, and caves which are challenging due to tough terrain and limited communication. Each system track team needed to deploy its multi-robot team supervised by a single human operator to locate and identify artifacts correctly with a 60-minute time limit. Team CSIRO Data61 used a heterogeneous multi-robot team (Fig. 7) including two BIA5 OzBot ATRs (tracked vehicles named Bear and Rat), two Boston Dynamics Spots (legged vehicles named Bingo and Bluey), and two Emesent Hovermap drones. Only the ground robots (ATRs and Spots) were controlled by the task-based approach, while the drones were launched by operator commands and independently explored afterwards.

Fig. 8 contains the timetable of each ground robot in the SubT final run, respectively. Each timetable illustrates the robot's current operation at each time step. During the final event, Bluey and Bingo (Spots) entered the environment first. To avoid Bluey (Spot) entering the dangerous area (like the railway tunnel where it had tripped in preliminary runs), it was controlled by the operator using waypoint commands periodically in the beginning. Then, both robots autonomously executed prioritized tasks in prioritization regions or tasks assigned to the robot by the operator, autonomously exploring distant parts of the course before falling outside the communication range. Rat (ATR) was mainly executing tasks autonomously until hardware failures. Bear (ATR) was controlled using waypoint commands after the communication with Bingo was lost around 20 minutes in. Since Bluey also lost communication later in the run, the operator used waypoint commands to guide Bear to act as a mobile relay point to attempt to regain communication with Bingo and Bluey while dropping communication nodes to maintain communication with the base station. Table I summarizes the distributions of how robots operated under various levels of control. These results shows that our interface provided the operator a clear overview of the situation to allow robots to run with a high level of autonomy, and enabled effective high-level control to guide the multi-robot team in a complex and challenging environment.

The graph-based prioritization region has proven to be a powerful tool for the operator to guide robots in a high-level interface. Since graph-based prioritization regions not only change the priorities of tasks within the regions, but also the tasks extended from the region based on the topometric graph, this makes it suitable for exploration where the boundaries are unknown. Fig. 9 provides a scenario showing how the operator used graph-based prioritization regions to guide the exploration of Bear (r3) to the area on the right, since the operator speculated that there was a large unexplored area. To do so, he created a low-prioritization region on the left to deprioritize exploration tasks that were extended to the left and a high-prioritization region on the right to prioritize exploration tasks that were extended to the right. As a result, task $EXPLORE(r3/1/130)$ was deprioritized and task $EXPLORE(r3/1/110)$ and $EXPLORE(r3/1/111)$ were prioritized. Bear (r3) executed $EXPLORE(r3/1/111)$ to keep exploring the area on the right. Later, Bear (r3) found a large cave area, which confirmed the operator's speculations. These results demonstrate that prioritization regions and interactive markers are effective for an operator to provide high-level control on the multi-robot team.

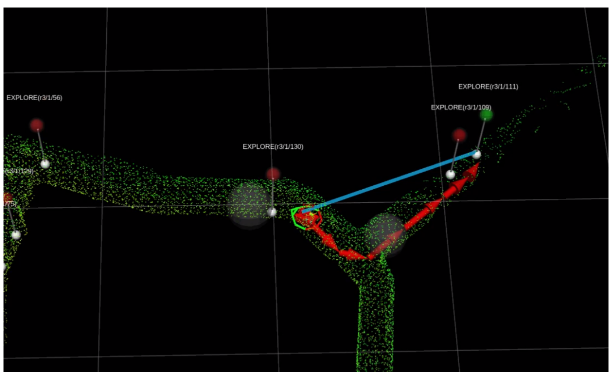


Fig. 9: A partial snapshot of the interface on how the operator created two graph-based prioritization regions to guide the exploration of Bear (r3) during the SubT Final Event. The operator created a low-prioritization region on the left and a high-prioritization region on the right so that exploration tasks on the right were prioritized.

B. Operator Feedback

After the DARPA Subterranean Challenge Final Event, the operator provided insightful feedback regarding the interface. Overall, the operator thought the interface was effective and did not feel limited when interacting with the robot at a high level. Specific feedback included:

- Because of the nature of the graph-based prioritization regions, they were effective for multi-robot exploration in an unknown environment of the SubT final event.
- The audio can convey precise information about which robot and whether it was a warning or an error, so that the operator can have a clear idea of the situation. More interestingly, since the previous missed message played when robots reconnected to the base station, the operator used audio cues as alerts for regaining communications, which turned out to be beneficial.
- The task interactive markers in the 3D map allowed direct control while providing full awareness about the states and locations of tasks.

V. Discussion

The design of user interface has evolved as our team learned more throughout the DARPA Subterranean (SubT) Challenge. In the earlier events (tunnel circuit in 2019), the control of the multi-robot team relied on waypoint command and teleoperation. However, this approach was challenging for the operator to maintain full situational awareness, especially when the environment was complex and the number of robots increased. In the following event (urban circuit in 2020), robots maintained a high level of autonomy during exploration without operator interventions. However, we learned that the human operator could provide useful instructions (e.g., interesting regions to explore) and low-level control in unexpected situations. As a result, we designed our multi-modal user interface to contain various levels of control from high-level guidance (default mode) to low-level teleoperation for different situations.

VI. Further Improvements

The multi-modal task-based graphical user interface has demonstrated its effectiveness on allowing an operator to control a multi-robot team in an unknown environment for time-sensitive missions. However, several areas of the interface could be further improved. First, the interface could have an optional high-detail task overlay to display information regarding why a task is selected or not: whether it causes conflict between robots; whether it is affected by any prioritization regions, and whether it is manually assigned by the operator. This overlay could explain task selections of robots, so the operator does not need to speculate why selections were made during the run. Second, the interface needs to have an explicit audio cue when a robot loses/regains communication. This has been shown to be critical for situational awareness in a communication-constrained environment. Third, since the operator sometimes fails to make the prioritization region fully cover the region he/she is interested in, robots may select tasks unintended by the operator. Therefore, the interface needs to have a more intuitive way of drawing 3D prioritization regions that can fully cover the regions and be able to adjust the prioritization regions after sending them to the robot.

VII. Conclusion

In this paper, we have presented a multi-modal task-based graphical user interface for controlling a multi-robot team in an unknown environment. The interface leverages audio cues and a map-based interface with task markers to provide the human operator with continuous situational awareness. Moreover, interactive task markers and prioritization regions provide the human operator with effective high-level control at both the single-task level and region (multi-task) level.

Team CSIRO Data61 has used the interface to prioritize regions for different robots for more efficient artifact searching in the DARPA SubT Challenge Final event and achieved second place, which shows the interface can aid the human operator on rapid decision-making effectively in time-critical scenarios. The designs of this interface can also extend to a wide-range of multi-robot applications.

REFERENCES

[1] "DARPA Subterranean Challenge." [Online]. Available: https://www.subtchallenge.com/

[2] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. L. Surdo, T. Molnar, R. Steindl, M. Wildie, I. Sa, N. Kottege, K. Stepanas, E. Hernandez, G. Catt, W. Docherty, B. Tidd, B. Tam, S. Murrell, M. Bessell, L. Hanson, L. Tychsen-Smith, H. Suzuki, L. Overs, F. Kendoul, G. Wagner, D. Palmer, P. Milani, M. O'Brien, S. Jiang, S. Chen, and R. C. Arkin, "Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61's approach to the darpa subterranean challenge," *Field Robotics*, vol. 2, pp. 595–636, 4 2021.

[3] H. L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, pp. 912–926, 2009.

[4] A. Rosenfeld, N. Agmon, O. Maksimov, and S. Kraus, "Intelligent agent supporting human–multi-robot team collaboration," *Artificial Intelligence*, vol. 252, pp. 211–231, 11 2017.

[5] J. Delmerico, S. Mintchev, A. Giusti, B. Gromov, K. Melo, T. Horvat, C. Cadena, M. Hutter, A. Ijspeert, D. Floreano, L. M. Gambardella, R. Siegwart, and D. Scaramuzza, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, pp. 1171–1191, 10 2019.

[6] P. Velagapudi, P. Scerri, K. Sycara, H. Wang, M. Lewis, and J. Wang, "Scaling effects in multi-robot control," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 2121–2126, 2008.

[7] B. Trouvain and H. L. Wolf, "Evaluation of multi-robot control and monitoring performance," 2002, pp. 111–116.

[8] G. Lunghi, R. Marin, M. D. Castro, A. Masi, and P. J. Sanz, "Multimodal human-robot interface for accessible remote robotic interventions in hazardous environments," *IEEE Access*, vol. 7, pp. 127 290–127 319, 2019.

[9] J. Scholtz, J. Young, J. L. Drury, and H. A. Yanco, "Evaluation of human-robot interaction awareness in search and rescue," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, pp. 2327–2332, 2004.

[10] A. Stoica, T. Theodoridis, H. Hu, K. McDonald-Maier, and D. F. Barrero, "Towards human-friendly efficient control of multi-robot teams," 2013, pp. 226–231.

[11] N. Chambers, J. Allen, L. Galescu, and H. Jung, "A dialogue-based approach to multi-robot team control," vol. 3. Springer, Dordrecht, 2005, pp. 257–262.

[12] S. Andolina and J. Forlizzi, "The design of interfaces for multi-robot path planning and control," vol. 2015-January. IEEE Computer Society, 1 2014, pp. 7–13.

[13] J. J. Roldán, E. Peña-Tapia, A. Martín-Barrio, M. A. Olivares-Méndez, J. D. Cerro, and A. Barrientos, "Multi-robot interfaces and operator situational awareness: Study of the impact of immersion and prediction," *Sensors 2017, Vol. 17, Page 1720*, vol. 17, p. 1720, 7 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/8/1720/htm https://www.mdpi.com/1424-8220/17/8/1720

[14] J. J. Roldán, E. Peña-Tapia, P. Garcia-Aunon, J. D. Cerro, and A. Barrientos, "Bringing adaptive and immersive interfaces to real-world multi-robot scenarios: Application to surveillance and intervention in infrastructures," *IEEE Access*, vol. 7, pp. 86 319–86 335, 2019.

[15] J. C. Scholtz, "Human-robot interactions: Creating synergistic cyber forces," 2002.

[16] B. Trouvain, C. Schlick, and M. Mevert, "Comparison of a map- vs. camera-based user interface in a multi-robot navigation task," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3224–3231, 2003.

[17] J. Crossman, R. Marinier, and E. B. Olson, "A hands-off, multi-robot display for communicating situation awareness to operators," *Proceedings of the 2012 International Conference on Collaboration Technologies and Systems, CTS 2012*, pp. 109–116, 2012.

[18] S. Mahadevan, "Visualization methods and user interface design guidelines for rapid decision making in complex multi-task time-critical environments," 2009.

[19] B. Trouvain and C. M. Schlick, "A comparative study of multi-modal displays for multirobot supervisory control," vol. 4562 LNAI. Springer, Berlin, Heidelberg, 2007, pp. 184–193.