International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:10, No:6, 2016

MIOM: A Mixed-Initiative Operational Model for Robots in Urban Search and Rescue

Mario Gianni, Federico Nardi, Federico Ferri, Filippo Cantucci, Manuel A. Ruiz Garcia, Karthik Pushparaj, Fiora Pirri

Abstract—In this paper, we describe a Mixed-Initiative Operational Model (MIOM) which directly intervenes on the state of the functionalities embedded into a robot for Urban Search&Rescue (USAR) domain applications. MIOM extends the reasoning capabilities of the vehicle, i.e. mapping, path planning, visual perception and trajectory tracking, with operator knowledge. Especially in USAR scenarios, this coupled initiative has the main advantage of enhancing the overall performance of a rescue mission. In-field experiments with rescue responders have been carried out to evaluate the effectiveness of this operational model.

Keywords—Actively articulated tracked vehicles, mixed-initiative planning interfeces, robot planning, urban search and rescue.

I. Introduction

URBAN Search & Rescue (USAR) robots play a crucial role in assisting first disaster responders [1]–[3]. Such robots can go where rescue workers are not allowed [2]. They can provide responders with a preliminary assessment of the conditions of the disaster area in which they are deployed [1].

In the last decade, robots participated in the rescue and recovery operations of many past and recent devastations, such as the 2001 World Trade Center (WTC) collapse [4] (see Fig. 1 (a)), the 2004 Mid Niigata earthquake in Japan [5], the 2005 Hurricanes Katrina, Rita, and Wilma in the United States [5] (see Fig. 1 (d)), the 2011 Tohoku earthquake and tsunami in Japan [6], [7] (see Fig. 1 (b)) and the 2012 Mirandola earthquake in the Emilia-Romagna region, Northern Italy [8], [9] (see Fig. 1 (c)).

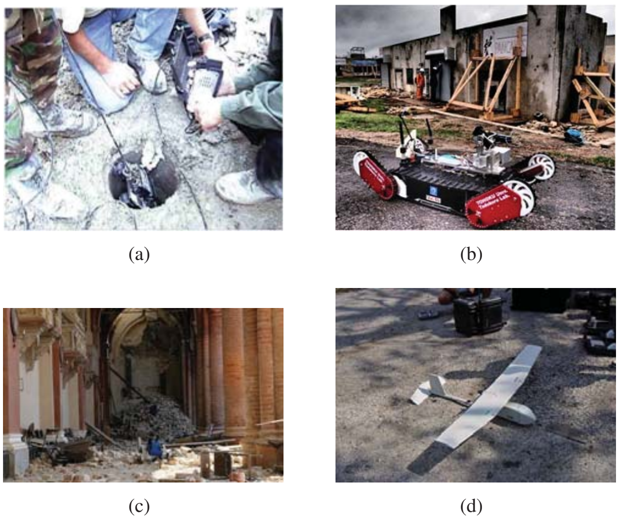

(a) (b) (c) (d)

Fig. 1 (a) A view of the Inuktun micro-VGTV robot being inserted into a sewer pipe at the World Trade Center site in an attempt to locate an entry to the basement; (b) Quince robot into the upper floors of the nuclear reactor building at the troubled Fukushima-Daiichi Nuclear Power Plant after tsunami waves disabled the reactors' power supply and cooling system. With radiation levels too high and dangerous for human workers, Quince was used to gather vital information; (c) Tracked vehicle Absolem assessing structural damages in San Francesco church, Mirandola, Italy; (d) Man-packable UAVs used to search portions of Mississippi during the hurricane Katrina response

Robots deployment in real rescue operations is nowadays possible due to the countless technological advances over the last decades in robotics aiming to improve the capabilities of these systems. References [5]–[7], [9], [10] have highlighted that one of the key attributes which makes robots effective for USAR domain applications is the semi-autonomous control. Semi-autonomous control is an operational mode which allows for task sharing between robots and operators [11]–[13]. Under this setting, the robot focuses on low level tasks whereas the human operator is in charge of high-level control and supervisory tasks [14]–[16]. However, although semi-autonomous control improved robot performance, enhancing both the operator situation awareness and human-robot interaction [4], [17], [18], it does not provide the operator with any form of interaction which also extends the reasoning capabilities of the robots. In large-scale environments the robot might not be able to autonomously find a path toward a target position due to either the high dimensionality of the searching space or the lack of information about the surrounding. Under semi-autonomous control, an operator can recover from such a stalemate only by manually operating the robot. Conversely, the operator might intervene directly on the path planning system of the robot by either limiting the area within which to search the path or filling the missing information with his/her knowledge. This initiative might effectively reduce the computational complexity of path planning.

In this paper, we describe a MIOM which enhances the reasoning capabilities of the robot, i.e. mapping, path planning, visual perception and trajectory tracking, with operator knowledge. Similar to a semi-autonomous control mode, MIOM incorporates operator activities into the control cycle involving both robot perception and execution. It interconnects human feedback with robot perception to build up a common ground for reasoning. Moreover, it constraints robot inference with human decisions.

The paper is organized as follows. The next section briefly

Mario Gianni is with the ALCOR, Vision, Perception and Learning Robotics Laboratory, DIAG, Sapienza University of Rome, Italy (e-mail: gianni@dis.uniroma1.it).

summarizes related work. Section III introduces MIOM. Section IV describes the development of the proposed operational mode on a real rescue robot. Finally, Section V describes the in-field experiments, with rescue responders, which have been carried out to evaluate the effectiveness of MIOM.

II. Related Work

A collaborative-shared control strategy that combines the operator abilities with robotic-based tasks has been proposed in [19]. In this strategy, the Collaborative Control component of the robot architecture is responsible for allowing operator intervention when the robot is facing complex situations, whilst the Shared Control component provides an automatic control mechanism to assist as well as to monitor-correct irrational operator actions. In [16], an approach for semi-autonomous navigation is proposed. This approach relies on the automatic detection of interesting navigational points and a human-robot dialog aimed at inferring the user's intended action. During this dialog, the user can confirm or reject the robots propositions making the system suitable for low throughput interaction devices, including brain-computer interfaces. Bruemmer and colleagues in [11] proposed a semi-autonomous controller which allowed an operator to set four different control modes: (1) Tele-mode; (2) Safe-mode; (3) Shared-mode and, (4) Auto-mode. In tele-mode, the operator manually controls the robot motion. In safe-mode, both navigation and object detection tasks are managed by the operator, while collision avoidance is performed by the robot. In shared-mode, the robot is responsible for generating optimal paths for navigation, given the target positions, instructed by the operator. In auto-mode, the operator is in charge of high-level task, such as defining a point to navigate to or searching a selected region, while the robot manages navigation and obstacle avoidance. The main drawback of the aforementioned approaches is that they do not allow the operator to change the level of autonomy of the robot on the fly, during a search and rescue operation [20]. An interesting approach to semi-autonomous control with dynamic adjustment of the level of autonomy of the robot has been described in [14], [21]. In this approach, the control system is responsible for coordinating the interventions of the human operator and the low level robot activities, under a mixed-initiative planning setting. The control system is endowed with a declarative model of the activities of the robot, specified in the Temporal Flexible Situation Calculus (TFSC) [22]. The model explicitly represents the main components and activities of the robot system, the cause-effect relationships as well as the temporal constraints among the activities. Further, the model integrates a representation of the activities of the human operator, enabling the control system to supervise the his/her operations. A reactive planning engine (i) monitors the consistency of the robot and operator activities, with respect to the model, managing failures and, (ii) incrementally generates plans, allowing the operator to locally assess the robot operations. The designed control schema endows the robot with some hybrid operative modalities lying between autonomous and teleoperated modes. Each of these mode is determined by the way in which the operator interacts with the control system. In [15] the authors described a sensor-based autonomous sub-track controller to manage swingable sub-tracks of rescue robots, which are used to negotiate steps and uneven terrain. The controller allows an operator to manually control the main tracks, while the autonomous controller is used for the sub-tracks. Despite the mentioned approaches to semi-autonomous control resulted to be very effective in minimizing the workload of the operator, they do not provide interfaces with the robot functionalities which allows operators to integrate knowledge aiming to also reduce computational payload.

III. The Mixed-Initiative Operational Model (MIOM)

Let us assume that a rescue robot is endowed with a set of functionalities. We denote with $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{P}$ the set of functionalities belonging to the perceptual, the control and planning system of the robot, respectively. We specify with $\mathbf{S}_{\mathcal{A}}$, $\mathbf{S}_{\mathcal{S}}$ and $\mathbf{S}_{\mathcal{P}}$ the sets of states of the functionalities in $\mathcal{S}$, $\mathcal{A}$ and in $\mathcal{P}$, respectively. Further, we assume that each of these sets is finite and that $\mathbf{S}_i \cap \mathbf{S}_j = \emptyset$, for $i, j \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$. For each set of functionalities, we define a finite set $\mathbf{T}_i$, with $i \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$, of transitions. We assume that $\mathbf{T}_i \cap \mathbf{T}_j = \emptyset$ for $i, j \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$. We denote with $\mathbf{E}_{\mathcal{S}}$, $\mathbf{E}_{\mathcal{A}}$ and $\mathbf{E}_{\mathcal{P}}$ the finite sets of internal events associated with each functionality in $\mathcal{S}$, $\mathcal{A}$ and in $\mathcal{P}$, respectively, with $\mathbf{E}_i \cap \mathbf{E}_j = \emptyset$ for $i, j \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$. Let us introduce a finite set of operator action $\mathbf{A}_i$ for each $i \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$. Here we assume that $\mathbf{A}_i \cap \mathbf{A}_j = \emptyset$ for each $i, j \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\}$. MIOM is defined by the following triple

$$\mathcal{M} = \langle \mathcal{M}_{\mathcal{S}}, \mathcal{M}_{\mathcal{A}}, \mathcal{M}_{\mathcal{P}} \rangle \tag{1}$$

with

$$\mathcal{M}_i = \langle \mathbf{S}_i, \mathbf{T}_i, \mathbf{E}_i, \mathbf{A}_i, \alpha_i, \beta_i, \lambda_i, \gamma_i, \delta_i \rangle \quad \text{with } i \in \{\mathcal{S}, \mathcal{A}, \mathcal{P}\} \tag{2}$$

Here $\alpha_i{:}\mathbf{T}_i \rightarrow \mathbf{S}_i$ is the mapping returning the source state $\alpha_i(t) \in \mathbf{S}_i$ of the transition $t \in \mathbf{T}_i$. $\beta_i{:}\mathbf{T}_i \rightarrow \mathbf{S}_i$ returns the target state $\beta_i(t) \in \mathbf{S}_i$. $\lambda_i{:}\mathbf{T}_i \rightarrow \mathbf{E}_i$ maps each transition $t \in \mathbf{T}_i$ to the internal event $\lambda_i(t) \in \mathbf{E}_i$ which triggers the transition. $\gamma_i{:}\mathbf{A}_i \rightarrow \mathbf{E}_i$ is a mapping from $\mathbf{A}_i$ to $\mathbf{E}_i$ which takes each operator action $a \in \mathbf{A}_i$ to an internal event $\gamma_i(a) \in \mathbf{E}_i$. Finally, $\delta_i{:}\mathbf{T}_i \rightarrow \mathbf{A}_i$ maps each transition $t \in \mathbf{T}_i$ to an operator action $\delta_i(t) \in \mathbf{A}_i$.

According to the model in (2), each $\mathcal{M}_i$ resembles a labeled transition system further extended with the two additional mapping $\gamma_i(\cdot)$ and $\delta_i(\cdot)$. These functions formally represent the intervention of the operator on the control and on the state of a robot functionality, respectively. Intuitively, $\gamma_i(\cdot)$ models the interaction between the robot and the operator as described in [14], [21] whilst $\gamma_i(\cdot)$ extends this operational mode with a form of interaction acting on the computational results of a robot functionality, that is, at reasoning level.

In the following, we describe, in more detail, the development of MIOM in a real rescue robotic system, better highlighting how operator intervention can affect the reasoning capabilities of the system in order to speed up computation.

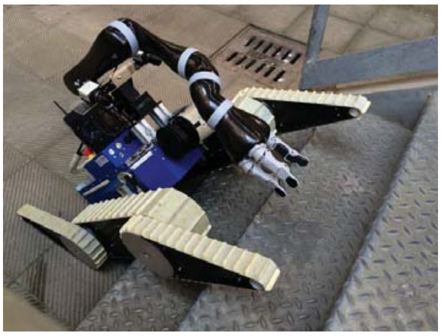


Fig. 2 Actively Articulated Tracked Vehicle Absolem, designed by ©Bluebotics for USAR applications. This platform is equipped with the KINOVA Jarm Arm for pick and place opeartions

IV. MIOM Development

Before describing a realization of MIOM, as defined in Section III, we have to introduce the description of the rescue robotic system in which MIOM has been grounded.

A. Robotic Platform

The robotic platform is an actively articulated tracked vehicle, initially designed by ©Bluebotics [23] and then, at a later stage, re-factored by Neovision s.r.o (see Fig. 2). This robot is endowed with a breakable passive differential system and two active sub-tracks, namely flippers, on both the ends of the tracks. The active flippers enhance climbing capabilities of the robot, while the mechanical differential increases the robot traction. The mechanical design maximizes the surface of the tracks in contact with the ground and improves the stability of such robot on stairs, ramps, rubbles and uneven terrain. The robotic platform is equipped with a rotating 2D SICK LMS-100 range finder, a LadyBug3 Omnidirectional camera and an XSens IMU/GPS inertial sensor unit. On top of the robot body, the lightweight 6DOFs Kinova Jaco Arm has been mounted for manipulation tasks. Moreover, a camera system on a Pan-Tilt Unit-D46-17, by FLIR, enhances terrain perception, morphological adaptation and navigation.

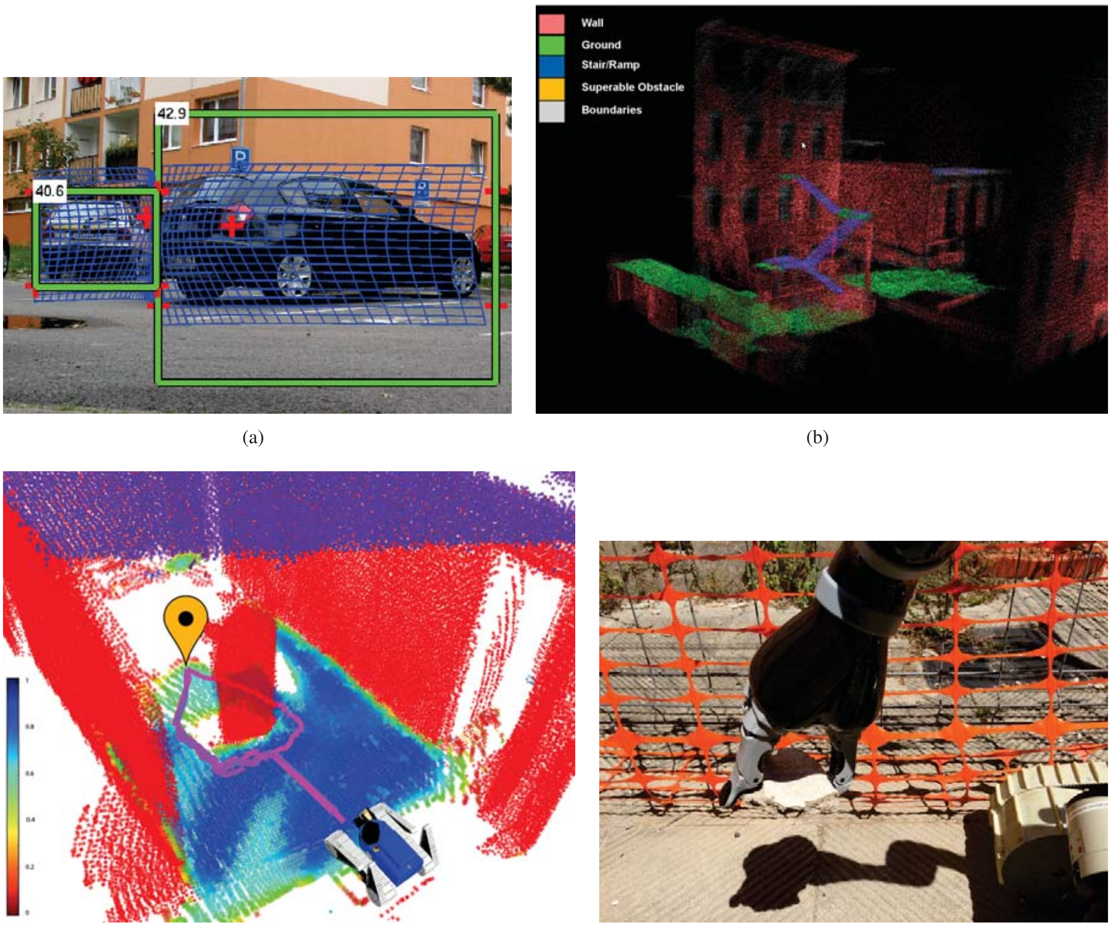
B. Robot Functionalities

The robot builds up a situation awareness of the environment, from raw data coming from the different sensors (see Fig. 3). This situation interpretation is based on a 3D metrical mapping of the environment [24]. In order to bridge the gap between robot-centric and human-centric situation awareness, this representation is extended with visual perception [25], [26], unsupervised and user-driven topological decomposition [27], [28], functional mapping [29], point cloud categorization, based on segmentation [30] and traversability analysis [31]. Situation awareness is further enriched with a representation of the radio signal strength supporting the operator to drive towards regions with good signal coverage so as to avoid dangerous connection drops [32]. The robot deploys on top of these representations reasoning and autonomous planning capabilities. These capabilities include morphological adaptation [33], trajectory planning and control [34], for complex terrain traversal tasks, and three different strategies for 3D path planning, based on the 3D map [35], the segmented map [30] and the traversability map representations of the environment [31], respectively. A real time algorithm for updating the 3D map with dynamic obstacles detection is further embedded within the robot in order to enhance both navigation and planning tasks [36]. Finally, the robot is equipped with baseline arm teleoperation implemented using the Kinova API [37]. The operator can control the arm with a gamepad in either a Cartesian coordinate system or the joint space. The user can take advantage of the robot camera sensors to have visual feedback on the relative position of the arm during teleoperation. The user has also the possibility to control the fingers to perform basic manipulation tasks (e.g. grasping and object). The functionality for the arm teleoperation also provides three additional basic features: (1) Automatic return to home position by pressing a button, (2) Safe emergency stop of the arm and (3) Estimation of the forces acting on the tool, based on motor current measurements.

C. MIOM Realization

MIOM realization is developed through a graphical interface extending the Operator Control Unit (OCU) of the robot. The extended OCU includes three graphical control elements which allow an operator to switch between the set of functionalities $\mathbf{S}_{\mathcal{A}}$, $\mathbf{S}_{\mathcal{S}}$ and $\mathbf{S}_{\mathcal{P}}$. The operator changes the set of functionalities of interest using tabs as navigational widgets (see Fig. 4). Each tab is composed of three main group frames. First frame provides a visual feedback of the current state $s{\in}\mathbf{S}_i$ of a functionality. Second frame is composed of a set of widgets (e.g., buttons, check boxes, sliders, combo boxes). Each widget is associated with an action $a{\in}\mathbf{A}_i$ which the operator can perform to change $s{\in}\mathbf{S}_i$. Finally, latter frame comprises the widgets labeled with the internal events $e{\in}\mathbf{E}_i$. Each widget in the second frames effectively realizes the mapping $\delta_i(\cdot)$ whilst each widget in the third frame implements the link between the operator actions and the internal events of a functionality, that is, the mapping $\gamma_i(\cdot)$. In the following, we describe the detailed implementation of the above skeleton for three robot functionalities, namely, traversability mapping [31], 3D path planning [30], [31], [35] and trajectory tracking [34], [38] so as to better clarify the realizations of both the mappings event-action $\gamma_i(\cdot)$ and transition-action $\delta_i(\cdot)$.

Traversability analysis builds on top of the 3D metrical map an estimate of the terrain characteristics related to traversability. This estimate is based on a point-wise cost function accounting for (1) terrain classification, (2) terrain roughness, (3) obstacle clearance and (4) point cloud density. Several processing phases of the incoming point cloud are required in the traversability mapping pipeline, such as point cloud segmentation based on normal estimation, estimation of the boundary regions for obstacle clearance and terrain roughness measurement from point cloud elevation. Therefore, the state of this functionality can be represented by the result



(a)

(b)

(c)

(d)

Fig. 3 (a) Car detection; (b) point cloud segmentation; (c) path planning on traversability mapping; (c) finger control of the arm

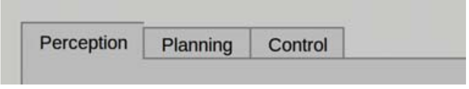


Fig. 4 Navigational widgets for switching between the set of functionalities

of each of these intermediate computational steps. Each of this result is visualized into a panel included in the state frame of the tab of the extended OCU associated with this functionality. As an example, traversability mapping is provided to the human operator in the form of a colored metric map. Colors range from blue to red according to estimated traversability, as illustrated in Fig. 3(c). A button has been added into the event frame according to the set of the internal events that we have identified for this functionality (see Fig. 5), namely

$$\mathbf{E}=\{\texttt{loadPCD}, \texttt{storePCD}, \texttt{computeNormals}, \texttt{segmentPCD}, \texttt{estimateRoughness}, \texttt{estimateBoundaries}\}$$

Computational cost of traversability analysis increases as the size of the incoming point cloud grows up. Moreover, errors in the estimation of point cloud normals, due to missing information of the surrounding, strongly affect the result of such an analysis. This is where the operator actions, introduced in (2) come in. Indeed, on the basis of the visual feedback, the operator might modify, on the fly, the estimated traversability cost of a region and then commits this state change to the robot. Traversability map can thus be updated with knowledge coming form an external supervisor so as to improve the estimates provided by the functionality. To this end, we have included in the operator action frame a set of buttons corresponding to the following set

$$\mathbf{A}=\{\texttt{selectPoint}, \texttt{selectPatch}, \texttt{modifyCost}, \texttt{Undo}, \texttt{Redo}, \texttt{Save}, \texttt{Load}, \texttt{Commit}\}$$

Here selectPoint allows the operator to modify the traversability estimates of a set of points in the selected one. selectPatch provides the operator with a handler through which he/she can manually circumscribe a set of points of a region and then change their cost value (see Fig. 6).

Three different algorithms can be used for 3D path planning. First algorithm performs D*-Lite searching [39] on a tensor map [35]. Second strategy performs searching on a weighted connectivity graph built on top of a semantic labeling of the 3D map [30]. Finally, the third algorithm combines together

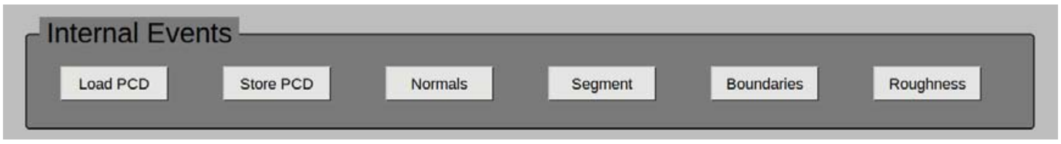
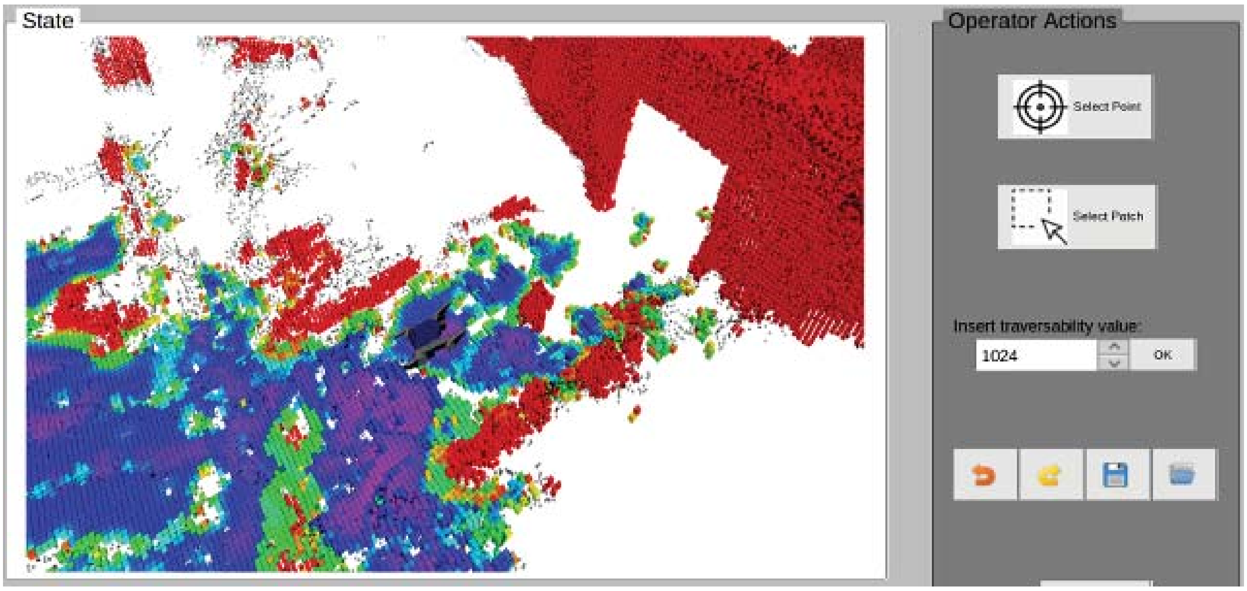


Fig. 5 Widgets for the set of internal events of traversability analysis



Insert traversability value: 1024

Fig. 6 Widget allowing an operator to manually circumscribe a set of points of a region to change traversability estimates

traversability analysis and a randomized statistical searching method for computing a path toward a goal position [31]. The state of each of these planning strategies, which is visualized into the state frame of the extended OCU, corresponds to the representation on top of which each strategy performs searching together with the associated size and the computed paths. The event frame includes a set of widgets (see Fig. 7) labeled with the names of the elements of the following set of internal events

$$\mathbf{E}=\{\texttt{setGoal},\texttt{chooseStrategy},\texttt{computePath},\texttt{Replan}\}$$

Missing information as well as the increasing size of the searching space, affecting computation, are the main causes of path planning failures. An approach for coping with these failures is to allow the operator to intervene on the planning cycle.

To this end, we identified two possible forms of interactions which can be implemented for supporting planning: (1) provide the operator with an interface for introducing way-points and (2) allows the operator to directly draw paths on top of the 3D map of the environment. First operator intervention has the main advantage of reducing the size of the searching space thus speeding up performance. Indeed, it constrains planning to perform searching on the portion of point cloud containing two selected way-points. On the other hand, the latter makes planning search complete. According to the above form of interactions, the set of operator actions of MIOM has been defined as follows

$$\mathbf{A}=\{\texttt{addWP},\texttt{removeWP},\texttt{moveWP},\texttt{commitWPS}\}\cup\\ \{\texttt{Trace},\texttt{Rollback},\texttt{Delete},\texttt{Smooth},\texttt{Commit}\}$$

An implementation of these actions in the extended OCU is illustrated in Fig. 8.

First subset of actions allows an operator to flexibly change the positions of the way-points. The interface for drawing paths is described in [38].

Robot control provides operators with several hybrid operative modalities ranging between autonomous and teleoperated modes, available during the execution of a task. Under teleoperation the operator manually controls all the degree of freedom of the robot. Under shared mode, the operator can manually control steering, while the robot is in charge of morphological adaptation or vice versa. The operator can switch between these operative modalities through a set of radio buttons suitably included in the control navigational widget. On the basis of the above modes, we defined the following set of events

$$\mathbf{E}=\{\texttt{moveForward},\texttt{moveLeft},\texttt{moveRight},\texttt{moveBack},\\ \texttt{turnLeft},\texttt{turnRight},\texttt{Stop}\}\cup\{\texttt{Flat},\texttt{Raise},\\ \texttt{Approach},\texttt{pushBack},\texttt{landDown},\texttt{landUp}\}$$

Each $e\in\mathbf{E}$ has been implemented by a widget in the internal event frame of the control tab of the extended OCU (see Fig. 9).

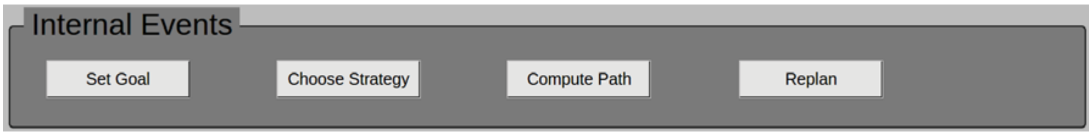


Fig. 7 Widgets associated with the internal events of planning

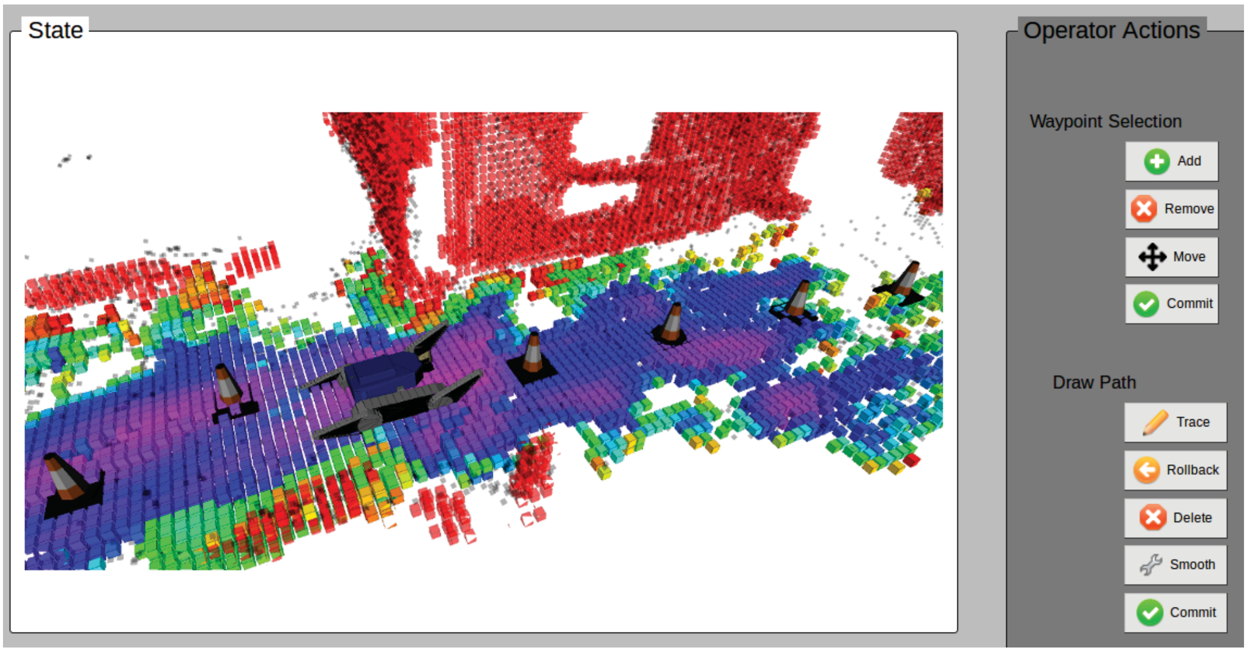


Fig. 8 Operator action widgets of path planning togheter with visual feedback of waypoints

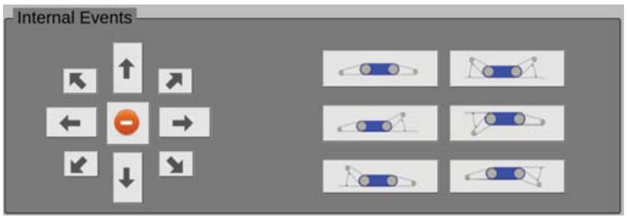
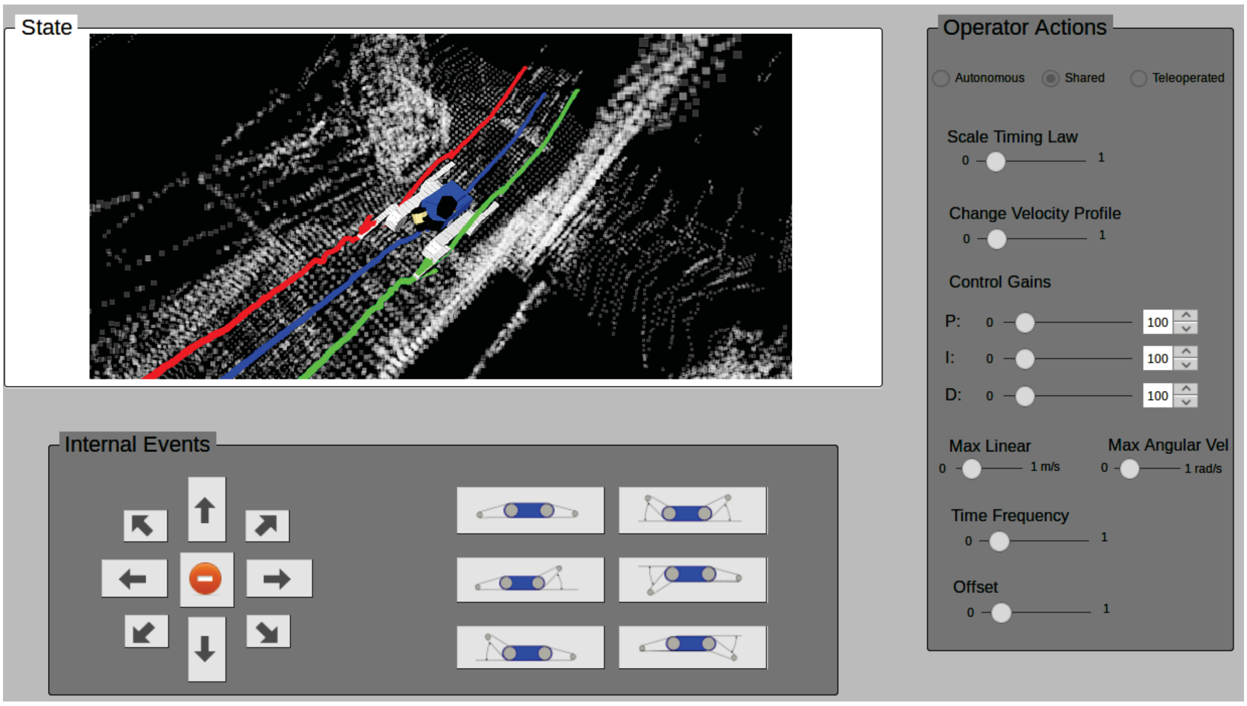


Fig. 9 Widgets in the internal event frame of the control tab of the extended OCU

Last subset of internal events corresponds to the minimal set of flippers configurations for executing a robot navigation task [35], especially stair-climbing (see Fig. 2). In particular, `pushBack` can recover the robot from flip-over, by raising down the platform when it is needed. `Raise` reduces both the effort of the flipper servo motors and the traction force on the robot body facilitating rotational motions [34]. Under autonomous mode the robot relies on a trajectory tracking controller in order to follow a route either computed by a path path planning strategy or provided by the operator through the interfaces, described above. The controller is also responsible for simultaneously adapting flippers posture to the terrain surfaces on which the paths lie [34], [38]. The operator can monitors, at real-time, the current state of execution, the trajectory tracking error and the controller parameters. This feedback is visualized within the state frame of the tab. However, due to slippage between the tracks and the terrain, the robot can significantly deviates from the planned trajectory hitting a nearby obstacle. Moreover, loss of contact between the tracks and the ridges of the stairs can occur during stair-climbing thus exposing the platform to serious risks. In order to limit such dangerous situations a solution might be to allow the operator to directly act on the internal setting of the controller. For example, by cutting the maximum speed of the controller, the operator can depress acceleration thus forcing the robot to move more slowly on terrains which are particularly harsh. With this in mind, we defined the following set of actions

$$\begin{aligned}\mathrm{A}=\{&\texttt{scaleTimingLaw}, \texttt{changeVelProfile},\\ &\texttt{setControlGains}, \texttt{setMaxLinVel}, \texttt{setMaxAngVel}\\ &\texttt{scaleTimeFreq}, \texttt{changeOffset}\}\end{aligned}$$

Note that for instance that by reducing maximum angular velocity through the action `setMaxAngVel`, the operator can regulates the oscillations of the heading direction of the robot during stair-climbing thus reducing both the lateral and longitudinal slippage between the tracks and the ridges of the stairs. Fig. 10 illustrates an implementation of these action within the control tab of the extended OCU.



○ Autonomous ● Shared ○ Teleoperated

Scale Timing Law: 0 – 1

Change Velocity Profile: 0 – 1

Control Gains
P: 0 – 100
I: 0 – 100
D: 0 – 100

Max Linear: 0 – 1 m/s
Max Angular Vel: 0 – 1 rad/s

Time Frequency: 0 – 1

Offset: 0 – 1

Fig. 10 Operator actions for changing the internal setting of the controller

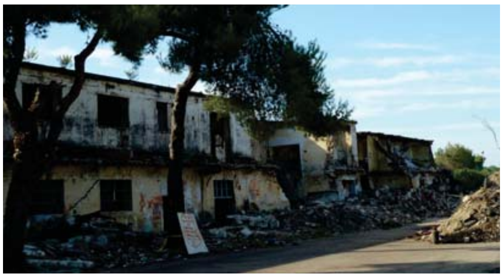


Fig. 11 Ex-American collapsed hospital, Calambrone, Italy

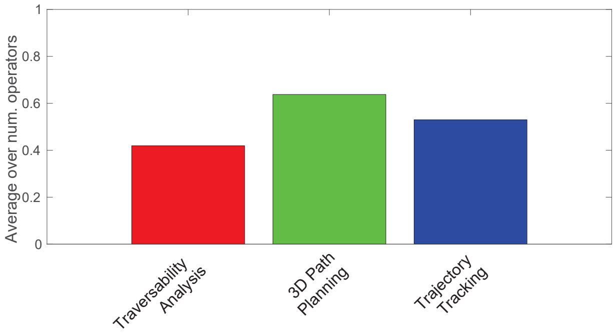


Fig. 12 Normalized histogram of the average number of times the tab for traversability, path planning and trajectory tracking, have been clicked, with respect to the number of operators

V. Experiments

In this section, we describe the experiments carried out to evaluate the benefits of the OCU, extended with the proposed realization of MIOM on the overall robot performance. The extended OCU has been implemented in RVIZ, the 3D visualizer of Robot Operating System (ROS), for displaying both robot sensor data and state [40].

10 skilled operators have been preliminary trained to use the functions implemented in each navigational widget. After this training phase, we deployed the tracked vehicle into a real rescue scenario (see Fig. 11). Within two rooms of the collapsed building we located dummies simulating victims. In addition, we randomly placed three signs for indicating the presence of dangerous sources.

We instructed the operators to perform a rescue mission with the tracked vehicle endowed with the extended OCU. This mission comprised searching for victims as well as the identification of the dangerous sources. For each operator we counted the number of times that navigational, internal event and action widgets have been used during the mission.

Fig. 12 shows the normalized histogram of the average number of times the navigational widget for traversability, path planning and trajectory tracking, have been clicked, with respect to the number of operators. From Fig. 12 it can be noted that, in average, both planning and execution required the support of the operators. Motivation is due to the harshness of the environment which made difficult for the robot to compute long-distance paths as well as to autonomously execute safe motions on the terrain surface.

Fig. 13 depicts the normalized histogram of the average number of times both internal event and action widgets of traversability analysis tab have been used, with respect to the number of operators. Here, we can observe that `estimateBoundaries`, `selectPatch` and `Undo` have been

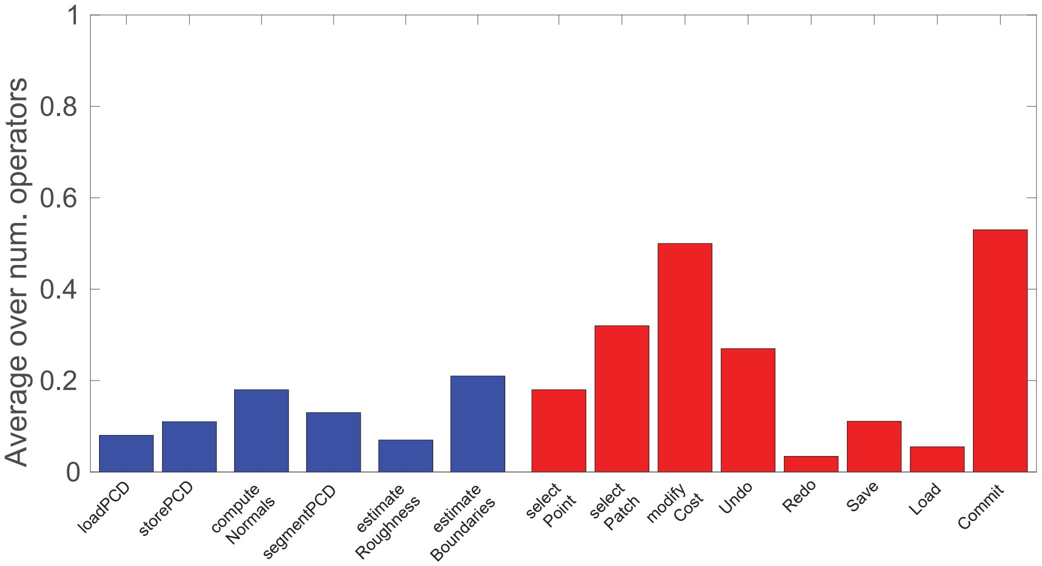


Fig. 13 Normalized histogram of the average number of times both internal event and action widgets of traversability analysis tab have been used, with respect to the number of operators

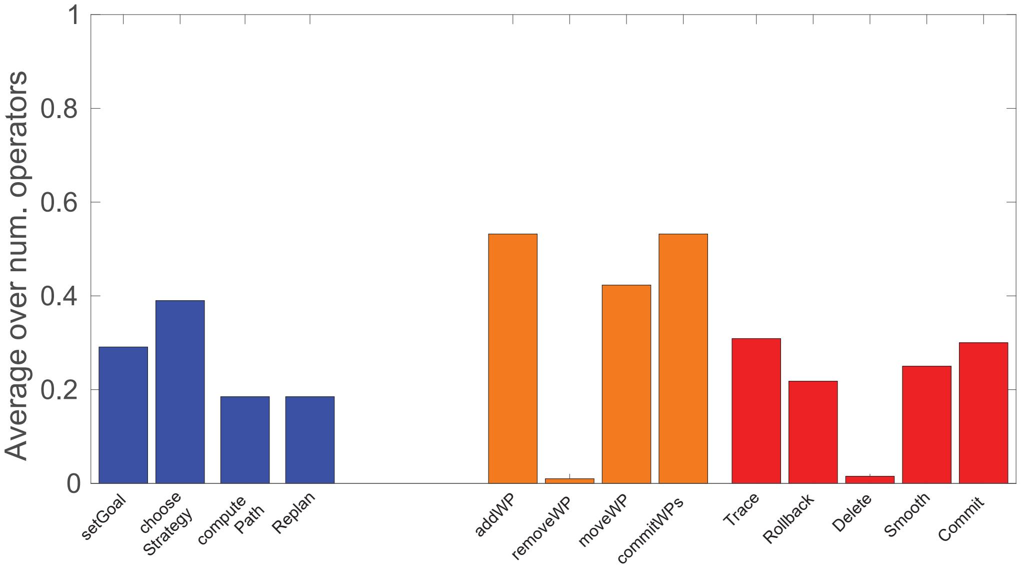


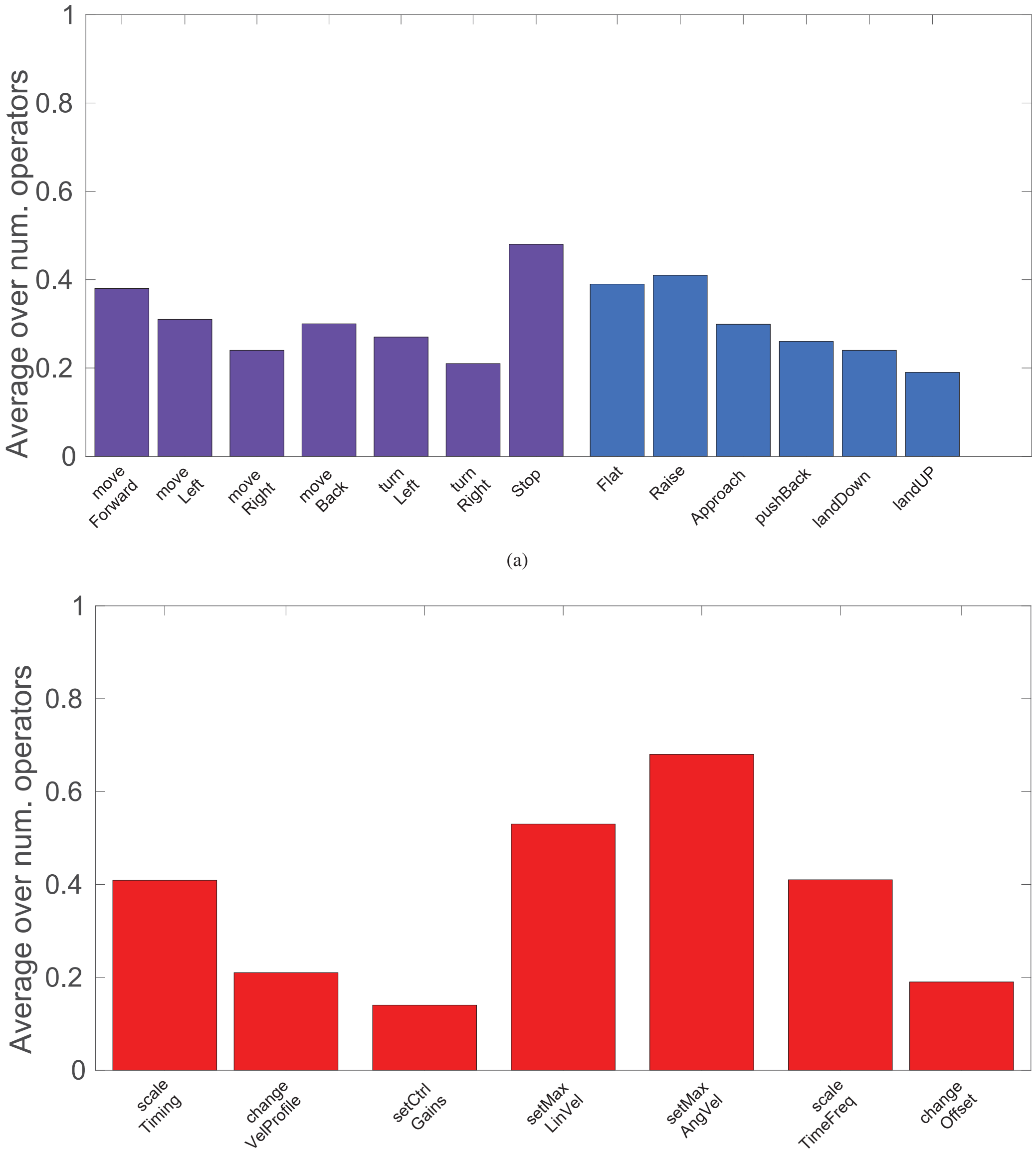
Fig. 14 Normalized histogram of the average number of times both internal event and action widgets of the path planning navigational tab have been used, with respect to the number of operators

selected more than others widgets. The main reason is that `estimateBoundaries` and `selectPatch` have been used for navigation and for filling missing data, respectively. `Undo` was due to operator mistakes. Last consideration suggests that this navigational widget has to include additional feedback for reducing such mistakes.

Fig. 14 reports the normalized histogram of the average number of times the operators interacted with both internal event and action widgets of the path planning navigational tab.

Results in Fig. 14 highlighted two different phases of the operators behaviors during the mission. In the first phase, operators tried manly rely on the planning strategies implemented on the robot. This explains the high values of the averages of the internal events. Operators alternate between `chooseStrategy` and `Replan` widgets. In the second stage, we have registered an intensive use of both `addWP` and `Trace`. This was due the continuous failures of the strategies in the first phase. These two actions of MIOM turned out to be more efficient than the internal events. Moreover, the degree of flexibility of MIOM significantly reduces operators mistakes. Indeed, values of both `removeWP` and `Delete` resulted to be very low (about 0.01 in average).

Fig. 15 (a) shows the normalized histogram of the average



(a)

(b)

Fig. 15 (a) Normalized histogram of the average number of times internal event widgets of control tab have been used, with respect to the number of operators; (b) normalized histogram of the average number of times operator actions have used.

number of times the operators used the internal event widgets of the control tab. On the other hand, Fig. 15 (b) shows the normalized histogram of the average number of times operator actions have used. Here, it can be noted from the results that operators assumed a careful behavior. Indeed, event widgets for stopping robot motions as well as for controlling flippers posture have been mostly clicked. Moreover, very often, the operators have forced the robot to move slowly on the traversed terrains. This behavior highlights that, despite the operational modes resulted to be very effective in minimizing the workload of the operator, they are still far from being really considered trustworthy by the operators [41].

Finally, Fig. 16 shows the trend of the computational cost, with respect to time, of path planning, with and without operator intervention through MIOM, as the size of the incoming point cloud increases. This trend demonstrates that both way-point selection and path drawing operator interactions speed up planning.

VI. Conclusion

In this paper, we described MIOM for robots in USAR. The main motivation behind this alternative model of interaction is to integrate operator knowledge within the reasoning capabilities of the robot in order to enhance both reliability and robustness of the system. We provided a formal definition, a skeleton and a realization of MIOM. Experiments under

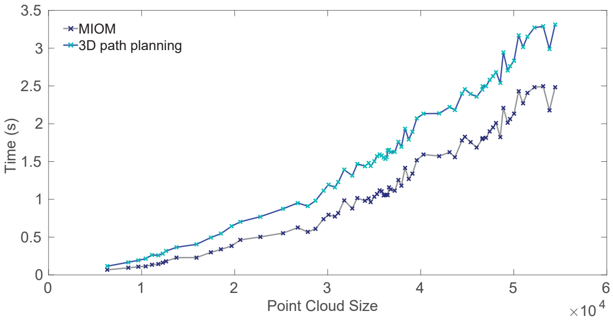


Fig. 16 Computational cost, with respect to time, of path planning, with and without operator intervention through MIOM, as the size of the incoming point cloud increases

human-robot interaction setting have proved the effectiveness of MIOM for building a more accurate representation of the environment, for speeding up path planning on harsh terrain and, finally, for supervising robot motions.

VII. Acknowledgment

This research is supported by the EU-FP7-ICT-Project TRADR 609763.

References

[1] J. Casper, M. Micire, and R. R. Murphy, "Issues in intelligent robots for search and rescue," in *Proc. SPIE*, vol. 4024, 2000, pp. 292–302.

[2] J. Casper and R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 3, pp. 367–385, 2003.

[3] J. L. Burke, R. R. Murphy, M. D. Coovert, and D. L. Riddle, "Moonlight in miami: A field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise," *Hum.-Comput. Interact.*, vol. 19, no. 1, pp. 85–116, 2004.

[4] R. Murphy, "Trial by fire [rescue robots]," *Robotics Automation Magazine, IEEE*, vol. 11, no. 3, pp. 50–61, 2004.

[5] R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. Erkmen, "Search and rescue robotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 1151–1173.

[6] E. Guizzo, "Japan earthquake: Robots help search for survivors," *IEEE Spectrum*, 2011, http://spectrum.ieee.org/automaton/robotics/industrial-robots/japan-earthquake-robots-help-search-for-survivors.

[7] E. Guizzo, "Japan earthquake: more robots to the rescue," *IEEE Spectrum*, 2011, http://spectrum.ieee.org/automaton/robotics/industrial-robots/japanearthquakemore-robots-to-the-rescue.

[8] G.-J. Kruijff, M. Janicek, S. Keshavdas, B. Larochelle, H. Zender, N. Smets, T. Mioch, M. Neerincx, J. van Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlavac, T. Svoboda, T. Petrcek, M. Reinstein, K. Zimmermann, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, P. Balmer, N. Tomatis, R. Worst, T. Linder, H. Surmann, V. Tretyakov, S. Corrao, S. Pratzler-Wanczura, and M. Sulk, "Experience in system design for human-robot teaming in urban search & rescue," in *Proceedings of 8th International Conference on Field and Service Robotics*, ser. STAR. Spring Verlag, 2012.

[9] G.-J. M. Kruijff, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, E. Pianese, S. Corrao, F. Priori, S. Febrini, S. Angeletti, V. Tretyakov, and T. Linder, "Rescue robots at earthquake-hit mirandola, italy: a field report," in *Proceedings of the 10th IEEE International Symposium of Safety Security and Rescue Robotics*, 2012, pp. 1–8.

[10] R. Murphy, "Trial by fire [rescue robots]," *IEEE Robotics Automation Magazine*, vol. 11, no. 3, pp. 50–61, 2004.

[11] D. Bruemmer, R. Boring, D. Few, J. Marble, and M. Walton, ""i call shotgun!": an evaluation of mixed-initiative control for novice users of a search and rescue robot," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 3, 2004, pp. 2847–2852.

[12] R. Wegner and J. Anderson, "Agent-based support for balancing teleoperation and autonomy in urban search and rescue," *Int. J. Robot. Autom.*, vol. 21, no. 2, pp. 120–128, 2006.

[13] B. Doroodgar, M. Ficocelli, B. Mobedi, and G. Nejat, "The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots," in *ICRA*, 2010, pp. 2858–2863.

[14] A. Finzi and A. Orlandini, "Human-robot interaction through mixed-initiative planning for rescue and search rovers," in *Proceedings of the 9th Conference on Advances in Artificial Intelligence*, ser. AI*IA'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 483–494.

[15] Y. Okada, K. Nagatani, K. Yoshida, T. Yoshida, and E. Koyanagi, "Shared autonomy system for tracked vehicles to traverse rough terrain based on continuous three-dimensional terrain scanning," in *IROS*, 2010, pp. 357–362.

[16] X. Perrin, R. Chavarriaga, F. Colas, R. Siegwart, and J. d. R. Millán, "Brain-coupled interaction for semi-autonomous navigation of an assistive robot," *Robot. Auton. Syst.*, vol. 58, no. 12, pp. 1246–1255, 2010.

[17] J. Drury, J. Scholtz, and H. Yanco, "Awareness in human-robot interactions," in *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 1, 2003, pp. 912–918.

[18] M. Baker, R. Casey, B. Keyes, and H. Yanco, "Improved interfaces for human-robot interaction in urban search and rescue," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 3, 2004, pp. 2960–2965.

[19] J. Shen, J. Ibanez-Guzman, T. C. Ng, and B.-S. Chew, "A collaborative-shared control system with safe obstacle avoidance capability," in *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, vol. 1, 2004, pp. 119–123.

[20] R. Murphy, , R. R. Murphy, and J. J. Sprouse, "Strategies for searching an area with semi-autonomous mobile robots," in *In Proceedings of Robotics for Challenging Environments*, 1996, pp. 15–21.

[21] A. Carbone, A. Finzi, A. Orlandini, and F. Pirri, "Model-based control architecture for attentive robots in rescue scenarios," *Autonomous Robots*, vol. 24, no. 1, pp. 87–120, 2008.

[22] A. Finzi and F. Pirri, "Representing flexible temporal behaviors in the situation calculus," in *IJCAI*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 436–441.

[23] Bluebotics, "Absolem surveillance & rescue," http://www.bluebotics.com/mobile-robotics/absolem/, 2011.

[24] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing icp variants on real-world data sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, April 2013.

[25] D. Hurych, K. Zimmermann, and T. Svoboda, "Fast learnable object tracking and detection in high-resolution omnidirectional images," in *VISAPP*, March 2011.

[26] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac, "Adaptive traversability of unknown complex terrain with obstacles for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, pp. 5177–5182.

[27] M. Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs," in *ICRA*, 2011, pp. 3269–3274.

[28] M. Liu, F. Colas, F. Pomerleau, and R. Siegwart, "A Markov semi-supervised clustering approach and its application in topological map extraction," in *IROS*, 2012, pp. 4743–4748.

[29] N. Goerke and S. Braun, "Building semantic annotated maps by mobile robots," in *TAROS*, 2009, pp. 149–156.

[30] M. Menna, M. Gianni, F. Ferri, and F. Pirri, "Real-time autonomous 3D navigation for tracked vehicles in rescue environments," in *IROS*, Chicago, USA, 2014.

[31] F. Ferri, M. Gianni, M. Menna, and F. Pirri, "Point cloud segmentation and 3D path planning for tracked vehicles in cluttered and dynamic environments," in *Proceedings of the 3rd IROS Workshop on Robots in Clutter: Perception and Interaction in Clutter*, Chicago, USA, 2014.

[32] S. Caccamo, R. Parasuraman, F. Baberg, and P. Ogren, "Extending a ugv teleoperation flc interface with wireless network connectivity information," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[33] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac, "Adaptive traversability of unknown complex terrain with obstacles for mobile robots," in *ICRA*, 2014.

[34] M. Gianni, F. Ferri, M. Menna, and F. Pirri, "Adaptive robust three-dimensional trajectory tracking for actively articulated tracked vehicles," *Journal of Field Robotics*, pp. n/a–n/a, 2015. [Online]. Available: http://dx.doi.org/10.1002/rob.21584

[35] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, and R. Siegwart, “3d path planning and execution for search and rescue ground robots,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, pp. 722–727.

[36] F. Ferri, M. Gianni, M. Menna, and F. Pirri, “Dynamic obstacles detection and 3d map updating,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[37] Kinova, “Kinova jaco arm api,” https://github.com/Kinovarobotics/kinova-ros, 2014.

[38] M. Gianni, G. Gonnelli, A. Sinha, M. Menna, and F. Pirri, “An augmented reality approach for trajectory planning and control of tracked vehicles in rescue environments,” in *Proceedings of the 11th IEEE International Symposium on Safety, Security and Rescue Robotics, Linkoping, Sweden*, 2013.

[39] S. Koenig and M. Likhachev, “D*lite,” in *Eighteenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 2002, pp. 476–483.

[40] M. Quigley, K. Conley, B. P. 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*, 2009.

[41] N. Gempton, S. Skalistis, J. Furness, S. Shaikh, and D. Petrovic, “Autonomous control in military logistics vehicles: Trust and safety analysis,” in *Engineering Psychology and Cognitive Ergonomics. Applications and Services*, ser. Lecture Notes in Computer Science, D. Harris, Ed. Springer Berlin Heidelberg, 2013, vol. 8020, pp. 253–262.