

STAGE: Scalable and Traversability-Aware Graph based Exploration Planner for Dynamically Varying Environments

Akash Patel, Mario A.V. Saucedo, Christoforos Kanellakis, and George Nikolakopoulos

Abstract—In this article, we propose a novel navigation framework that leverages a two layered graph representation of the environment for efficient large-scale exploration, while it integrates a novel uncertainty awareness scheme to handle dynamic scene changes in previously explored areas. The framework is structured around a novel goal oriented graph representation, that consists of, i) the local sub-graph and ii) the global graph layer respectively. The local sub-graphs encode local volumetric gain locations as frontiers, based on the direct pointcloud visibility, allowing fast graph building and path planning. Additionally, the global graph is built in an efficient way, using node-edge information exchange only on overlapping regions of sequential sub-graphs. Different from the state-of-the-art graph based exploration methods, the proposed approach efficiently re-uses sub-graphs built in previous iterations to construct the global navigation layer. Another merit of the proposed scheme is the ability to handle scene changes (e.g. blocked pathways), adaptively updating the obstructed part of the global graph from traversable to not-traversable. This operation involved oriented sample space of a path segment in the global graph layer, while removing the respective edges from connected nodes of the global graph in cases of obstructions. As such, the exploration behavior is directing the robot to follow another route in the global re-positioning phase through path-way updates in the global graph. Finally, we showcase the performance of the method both in simulation runs as well as deployed in real-world scene involving a legged robot carrying camera and lidar sensor.

Index Terms—Autonomous navigation, GPS-denied environments, exploration, dynamic environments, aerial and legged robots.

I. INTRODUCTION

While the exploration capabilities of autonomous robots have been increasingly improving in recent years through novel navigation approaches, the design of exploration missions is underutilized in terms of environment representation. The majority of the exploration algorithms are developed with the intent to increase exploration metrics by remodeling exploration goal and/or path selection process. Traditional frontiers [1] or sampling [2] based exploration approaches have proven to perform well in small scale, less complex environments. However, in subterranean environments such as underground corridors, tunnels, mines and caves, there exist additional factors such as degraded perception, complex topology, multi-branched self-similar structures, dynamic

The authors are with Robotics & AI Team, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden.

Corresponding author: akapat@ltu.se

The evaluation of stage planner is presented at https://youtu.be/NC4r_hhfZ-M?si=cFwbdwqdlFaRnll_

This work has been partially funded by the Sustainable Underground Mining Academy SP14 project.

obstacles that challenge autonomous exploration and mapping operations of robots. The DARPA organized Subterranean Challenge [3], [4], [5], [6] aimed to contribute in advancing the capabilities of autonomous robots in real life underground environments for Search And Rescue (SAR) missions. Autonomous deployment of robots in scenarios such as SAR missions, require the navigation autonomy to consider efficient exploration along with appropriate representation of the acquired knowledge of the explored environment. The dynamic change in the SubT environment also seen in the DARPA Subterranean challenge was a major obstruction for autonomous robots repositioning to a frontier in exploration. As part of development efforts and lessons-learned from DARPA SubT challenge of team COSTAR, we propose the Scalable and Traversability-Aware Graph based Exploration (STAGE) planner, that targets exploration proposing a memory efficient mechanism for fast local sub-graphs generation and minimal information exchange (collision checks on overlapping sub-graphs) to update the global graph. The merit of this representation of the environment allows for dynamical updates of the global graph in situations of scene changes (e.g. blocked path). As such, to the best of the authors knowledge STAGE enables long-term autonomous robot navigation in GPS-denied and dynamically varying environments.

II. RELATED WORKS

Building upon the original idea of frontiers in robotics context, several algorithms [7] present different exploration approaches by either selecting a candidate frontier or selecting a path to candidate frontier to demonstrate efficient exploration using frontiers based method. Apart from frontiers, there exist also sampling based approached where randomly sampled nodes are connected to form a tree structure spanning from robot's current position as root. Such approaches are generally classified as variant of Rapidly exploring Random Trees or RRT [8] that consider associated information gain with the nodes to compute local exploration goal. Both frontiers and sampling based approaches have their advantages and disadvantages based on the chosen environment. The frontiers based approach focuses on rapid exploration of unknown environments [9], [10] but suffer from partially exploring small areas in the map. On the other hand, the sampling based [11] approaches tend to efficiently cover local areas while overall performance is degraded on large scale environments. There exist also hybrid approaches [12] that combine both frontiers and Next Best View legacy to form an exploration approach that rapidly explores small scale unknown areas.

As part of development efforts within the DARPA SubT challenge, the competing teams demonstrated legged and aerial robot exploration through approaches tailored to fit subterranean settings. Most teams utilized local and global exploration planners where local planner drives the exploration through frontiers and/or RRT by computing A* paths to goals and utilize full explored map to plan computation heavy global or homing path in case of dead end or exceeding robot endurance limits [13]. The authors in [14] [GB planner] proposed a novel graph based exploration planner that constructs a graph through random sampling in adaptive oriented bounding box for local RRT-style exploration while building a global graph from previously visited nodes to plan global re-positioning and homing paths. The experimental evaluation of approach in [15] has proven to be the state-of-the-art in terms of adaptive exploration in unknown SubT environments. However, the GB planner still has limitations in terms of A) increasing path computation time on large scale exploration missions due to single layer navigation graph and B) Unable to adapt to change in the environment where global graph is already built. Through the capabilities of the proposed multi layer graph representation, traversability factor based adaptive path improvement and efficient planning, the STAGE planner aims to address open challenges in autonomous navigation in SubT environments.

III. CONTRIBUTIONS

Based on the aforementioned state-of-the-art works in the field of subterranean exploration, the key contributions and highlights of the STAGE planner are mentioned below.

The first contribution stems from the novel goal oriented graph generation that distinguishes STAGE planner from classical sampling based exploration approaches. The STAGE planner utilizes direct point cloud visibility to extract constantly updating frontier nodes to adaptively tune the local bounds for rapid graph building and graph improvement. Based on the frontier goals, an adaptively tuned sub-graph is generated for navigating the robot towards local volumetric gain-rich area. Furthermore, the proposed graph navigation scheme *re-utilizes* previously built sub-graphs to save additional computation at global graph building stage. The STAGE planner extracts overlapping regions from previously built sub-graphs and performs collision checks *only on the overlapping regions* to stitch the graphs together forming a dense global graph.

The second contribution of the STAGE planner is derived to address the particular challenge of how the robot can adapt to unknown change in the global map while re-visiting previously visited places in the global re-positioning or homing manoeuvres. Nearly all methods in literature adapt either back tracking on previously visited positions or utilizing a global graph for navigation to global re-positioning frontier. In real life SAR missions, the environment may change during the mission and the global graph built in previously seen as *Traversable* zone may be in collision with obstacle or occupied zones. To address this, we propose iterative update on execution path built from path ways of global graph. In

contrast to blindly following the global path, the STAGE planner segments the global path into smaller segments with size proportional to the current point cloud visibility. Reusing the vertices of the segment of the path and sampling sparse vertices around the segment to assure a traversable path from robot current state to the end of the current segment. If such segment is found to be in untraversable zone due to the unknown change in the environment, the planner discards the current path and re-computes global path ways by taking into account the change in the global map.

IV. STAGE PLANNER ARCHITECTURE

The exploration of unknown subterranean environments is a problem broadly studied during the DARPA SubT challenge enabling the robotized autonomy for exploring and mapping the unknown underground terrains. The two major autonomy challenges observed during the competition were, A) Efficient exploration-path planning in varying topology of SubT settings, B) Global re-positioning or homing path planning in presence of unknown change in the environment. Let us denote the current point cloud scan as \mathcal{P} , the local 3D occupancy map as \mathbb{M}_L , explored global 3D map as \mathbb{M}_G and set of potential frontier nodes as \mathcal{F} . where, $\mathcal{P} = \{(x_i, y_i, z_i) \mid i \in N\}$. The occupancy maps are often constructed with octree data structure where, a voxel v with three possible states v_f , v_o and v_u (corresponding to occupancy probability) is subdivided into eight voxels until a minimum volume defined as the resolution of the octree v_{res} is reached. From the current point cloud \mathcal{P} and local occupancy map \mathbb{M}_L , ($\mathbb{M}_L \subset \mathbb{M}_G$) the set of frontiers \mathcal{F} with at least k number of unknown neighbouring voxels need to be extracted and inserted into \mathbb{M}_G . Based on the unknown neighbours density of a frontier voxel, the volumetric exploration gain V_{gain} is computed. In order to perform rapid local exploration, the current visibility of the robot surroundings is depicted through local point cloud scan to orient the exploration direction in volumetric gain rich area. The navigation from the robot's current state $\zeta_i = \{(x_i, y_i, z_i, \psi_i) \mid i \in N\}$ to a potential frontier, it builds a traversable sub-graph \mathbb{S}_G^i composed of nodes $\mathcal{N} = \{n_1, n_2, \dots, n_n\}$ and edges $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$. A local exploration path λ_L to a potential frontier in field of view is chosen among the local path ways P_L to all local frontiers. Throughout the paper, var_i denotes the value of variable var at i^{th} planning iteration. For global navigation in known map, a dense global graph \mathbb{G}_G is built to find multiple paths to multiple global frontiers as well as the location of the mission base station. Subject to the robot's latest knowledge of the global occupancy map \mathbb{M}_G , the robot queries path-ways to all potential global-repositioning candidates \mathcal{C} where, $\mathcal{C} \subset \mathcal{F}$. Based on the volumetric gain evaluation of the re-positioning target frontier candidate and the information gain I along the path, a global re-positioning path-way λ_G needs to be computed to assure continued exploration. In contrast to blindly following the path λ_G , the important aspect of adaptive global re-positioning is to continuously assess the uncertainties Υ_u while navigating along pre-computed λ_G to assure collision free path execution step.

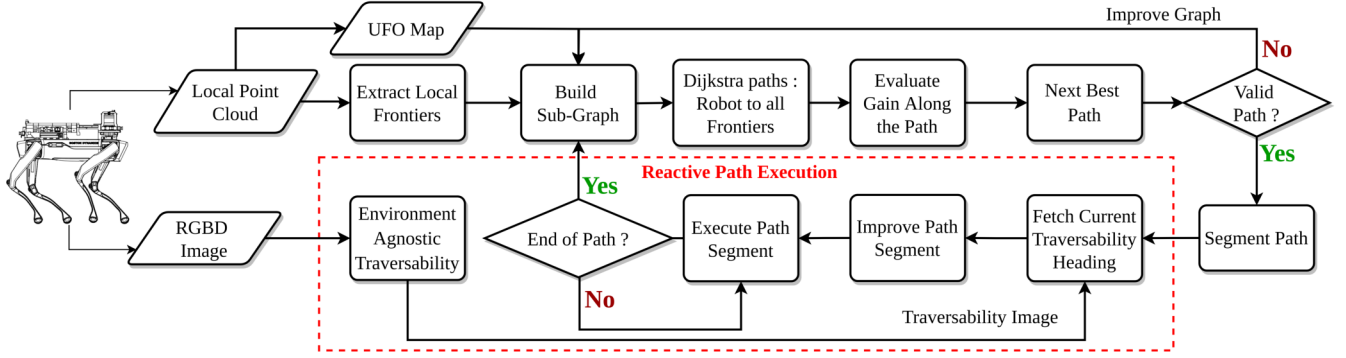


Fig. 1: Overview of the STAGE Local Exploration Planning scheme.

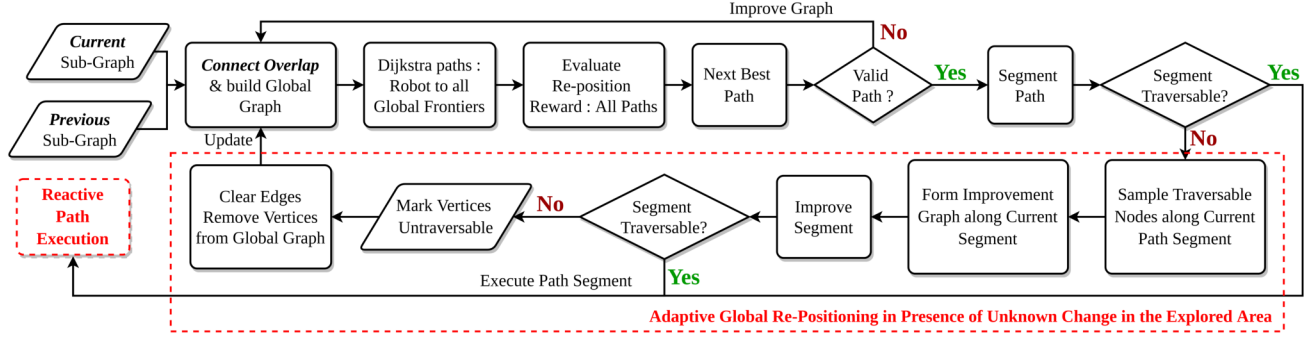


Fig. 2: Overview of the STAGE Global Re-positioning Planning scheme.

A. Local Exploration

As depicted in Fig. 2, the inputs to the STAGE local exploration stage are local point cloud scan \mathcal{P} and robot current state ζ . Local map \mathbb{M}_L update iterates through only the \mathbb{M}_L to fetch new frontier voxels. Based on \mathcal{P} , a direct Pointcloud visibility based Planning space ($P_v P_s$) is tuned to iterate through local UFOMap [16] in order to sample free nodes \mathcal{N} surrounding the robot. Taking into consideration the limited locomotion of legged robot, the nodes are only sampled in 2.5D allowing small elevation change from ζ_z for legged robot and in 3D for an aerial robot. Further considering the type of robot as input parameter, the planner forms connections between n and $n_{neighbour}$ where node $n \in \mathcal{N}$. A sub-graph \mathbb{S}_G for local navigation is built with \mathcal{N} and \mathcal{E} . In order to reduce computational overhead, the nodes in \mathcal{N} are examined with *kdtree* data structure to find if a node has minimum required neighbouring voxels that are frontiers. Using the Dijkstra shortest path [17], shortest paths to all frontier nodes in sub-graph are computed.

$$\mathbb{R} = \underbrace{W_i * \log\left(\frac{e^{W_u * v_u} + e^{W_f * v_f}}{e^{W_o * v_o}}\right)}_{\text{Volumetric gain}} - \underbrace{W_h * \left(\tan^{-1}\left(\frac{\zeta_y - n_y}{\zeta_x - n_x}\right) - \zeta_\psi\right)}_{\text{Heading penalty}} \quad (1)$$

To further assure local rapid exploration, we constrain the robot's field view to narrow down search for informative frontier nodes. A local exploration reward \mathbb{R} is computed

through (1) that takes into account the volumetric gain associated with the node in sub-graph and deviation from the robot's current heading direction ζ_ψ . In (1), W_i, W_u, W_f, W_o and W_h are the weights associated with information gain, unknown, free, occupied and heading deviation. Among the local path ways P_L a local exploration path λ_L to a node with highest local exploration reward \mathbb{R} is computed. Since the planning is done through collision checks based on ray casting, STAGE planner introduces a reactive path execution step to assure robot-safe path. The path λ_L is further segmented and the current segment in the path λ_L^k is refined to adapt the traversability information. The end point of path segment λ_L^k is shifted to the most traversable pixel in the Semantic Traversability image ($\mathbb{S}\mathbb{T}_{image}$) as shown in Fig. 5. Using the local exploration algorithm proposed in algorithm 1, in an iterative manner local exploration is continued until there is no new local information gain associated to frontiers in field of view. At this moment, a global re-positioning path is requested from global re-positioning planner to relocate the robot to a potential frontier of \mathbb{M}_G .

B. Global Re-Positioning Planning

As mentioned in subsection IV-A the global re-positioning to potential frontier is triggered when local exploration is insufficient. For global navigation the novelty of the STAGE planner stems from the re-utilization of previously built sub-graphs \mathbb{S}_G^i & \mathbb{S}_G^{i-1} to build a dense global graph \mathbb{G}_G . To connect the sub-graphs, we first detect the overlap in tuned planning spaces $P_v P_s$ corresponding to the latest and previous

Algorithm 1: Local Exploration

Input: \mathcal{P}, ζ

- 1 $\mathcal{F} \leftarrow \text{DetectFrontiers}(\mathcal{P}, \mathbb{M}_L)$
- 2 $\mathbb{M}_G \leftarrow \text{RegisterGlobalFrontiers}(\mathcal{F})$
- 3 **if** $\mathcal{F} \neq \emptyset$ **then**
- 4 $P_v P_s \leftarrow \text{TunePlanningSpace}(\zeta, \mathcal{P})$
- 5 $\mathcal{N} \leftarrow \text{SampleNodes}(P_v P_s)$
- 6 $\mathcal{E} \leftarrow \text{FormConnections}(\mathcal{N}, \mathbb{M}_L)$
- 7 $\mathbb{S}_G^i \leftarrow \text{BuildSubGraph}(\mathcal{N}, \mathcal{E})$
- 8 **for** *node* $n : \mathbb{S}_G^i$ **do**
- 9 $\text{NearestNeighboursLookup}(\mathcal{F})$
- 10 **if** $\text{neighbouring_frontiers} > k_{\text{threshold}}^{nn}$ **then**
- 11 $n \leftarrow \text{is_frontier}$
- 12 $n \leftarrow \text{ComputeVolumetricGain}(\mathbb{M}_L)$
- 13 $P_L \leftarrow \text{DijkstraPathsToFrontierNodes}(\mathbb{S}_G)$
- 14 $\lambda_L \leftarrow \text{ComputeCandidatePath}(P_L)$
- 15 $\lambda_L^k \leftarrow \text{SegmentPath}(\lambda_L)$
- 16 $\lambda_{\text{refined}}^k \leftarrow \text{RefinePathSegment}(\text{ST}_{\text{image}})$
- 17 $\text{nMPC} \leftarrow \text{ExecutePathSegment}(\lambda_{\text{refined}}^k)$
- 18 $\mathbb{G}_G \leftarrow \text{UpdateGlobalGraph}(\mathbb{S}_G^i)$
- 19 **else**
- 20 **RunGlobalRepositioningPlanner**

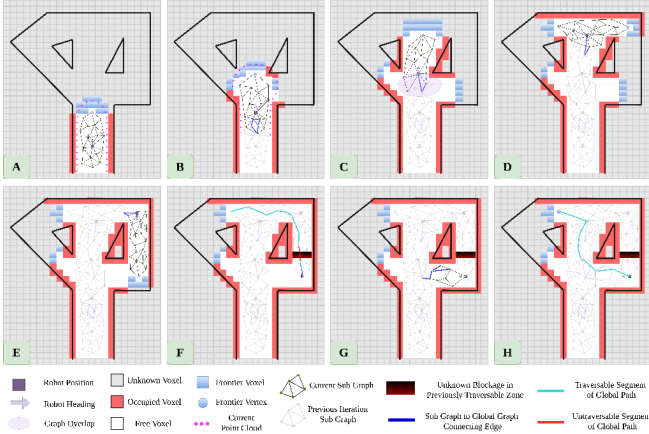


Fig. 3: STAGE planner overview visualization in A-H. Steps A-E depict the sequential global graph build through the association of the local sub-graphs. Step F visualizes the concept of obstructed path in the global graph during global re-positioning and the removal of connected nodes of the global graph in the untraversable area. Step G visualizes the sub-graph on an unexplored nearby alternative area to reach the re-positioning global frontier. Step H depicts the global path-way segments on the adapted global graph.

iteration sub-graphs. As shown in algorithm 2 the nodes belonging to the overlapping region are connected with the neighbours from the previous sub-graph to form continuous connection in \mathbb{G}_G from robot's current position to the location of the mission base station. A visual representation of the graph overlap and global graph building step is shown in Fig. 3. Through planning steps A to E in Fig. 3, the robot builds \mathbb{G}_G by re-utilizing \mathbb{S}_G . Through Dijkstra shortest paths algorithm, we further calculate path ways leading to all potential global frontiers from robot's current state. The reward \mathbb{R} is computed for each node along the global path way $\lambda_G \in P_G$ to compute global re-position reward for $\lambda_{\mathbb{R}}$. As described in (2) the re-position reward is further penalized through path length d_λ where, W_d is the weight associated

Algorithm 2: Global Re-positioning

Input: $\mathbb{S}_G^i, \mathbb{S}_G^{i-1}, \zeta$

- 1 $\mathcal{F} \leftarrow \text{DetectFrontiers}(\mathcal{P}, \mathbb{M}_L)$
- 2 **if** $\mathcal{F} \neq \emptyset$ **then**
- 3 $\mathbb{M}_G \leftarrow \text{RegisterGlobalFrontiers}(\mathcal{F})$
- 4 **if** UpdateGlobalGraph **then**
- 5 $\mathbb{G}_G \leftarrow \text{InsertSubGraphs}(\mathbb{S}_G^i, \mathbb{S}_G^{i-1})$
- 6 $\mathcal{N}_{\text{overlap}} \leftarrow \text{DetectOverlap}(\mathbb{S}_G^i, \mathbb{S}_G^{i-1})$
- 7 $\mathbb{G}_G \leftarrow \text{FormConnections}(\mathcal{N}_{\text{overlap}})$
- 8 $P_G \leftarrow \text{DijkstraPathsToFrontierNodes}(\mathbb{G}_G)$
- 9 $\lambda_G \leftarrow \text{EvaluateGlobalPathways}(P_G)$
- 10 $\lambda_G^k \leftarrow \text{SegmentPath}(\lambda_G)$
- 11 EvaluateSegmentTraversability($\mathbb{M}_L, \lambda_G^k$)
- 12 **if** $\lambda_G^k \text{ is UNTRAVERSABLE}$ **then**
- 13 $O_{ss} \leftarrow \text{OrientedSamplingSpace}(\zeta, \lambda_G^k)$
- 14 SampleNodes(O_{ss})
- 15 FindAlternativeSegment()
- 16 **if** !alt_segment_traversable **then**
- 17 $P_G \leftarrow \text{RemovePathWay}(\lambda_G)$
- 18 $\mathbb{G}_G \leftarrow \text{UpdateGraph}(P_G)$
- 19 $\lambda_G \leftarrow \text{EvaluateGlobalPathways}(P_G)$
- 20 $\lambda_G \leftarrow \text{RePlan}(\mathbb{G}_G)$
- 21 $\lambda_{\text{refined}}^k \leftarrow \text{RefinePathSegment}(\text{ST}_{\text{image}})$
- 22 EvaluateSegmentTraversability($\mathbb{M}_L, \lambda_{\text{refined}}^k$)
- 23 $\text{nMPC} \leftarrow \text{ExecutePathSegment}(\lambda_{\text{refined}}^k)$
- 24 **else**
- 25 **PlanReturnToHome**()

with path length penalty.

$$\lambda_{\mathbb{R}} = \sum_{p=1}^p \mathbb{R}_p - W_d * d_\lambda \quad (2)$$

1) *Uncertainty Aware Re-Planning:* As motivated above, the STAGE planner aims to handle dynamic changes in the explored map with the merit to fulfil global re-positioning without utilizing full \mathbb{M}_G during path re-planning. Dynamic changes of the scene can occur in such types of environments including rockfalls, blocked passages due to machine breakdowns, doors and other types of pathway obstructions. As it might happen during a change of the scene, the section of the global map $\mathbb{M}_\delta \subset \mathbb{M}_G$ is changed after the robot had initially mapped the \mathbb{M}_δ . Since previously it had traversed through \mathbb{M}_δ , the robot had built the global graph where the global path ways P_G pass through \mathbb{M}_δ . If an unknown uncertainty Υ_u is introduced in the \mathbb{M}_δ at a time instance that the robot is not present in the scene, the planned global pathways P_G will be in collision with the unknown Υ_u . If the robot blindly follows a path way $\lambda_G \in P_G$, it will result in collision with the unknown change in the global map as shown in planning step F of Fig. 3.

To address this, we introduce adaptive graph updates without utilizing full \mathbb{M}_G . As depicted in algorithm 2, before the path segment execution, the planner constructs an Oriented Sampling Space O_{ss} oriented from ζ towards the end point of current path segment λ^k . A ray casting operation is performed from ζ along λ^k . If this ray casting reports collision with occupied or unknown voxel, intuitively it is assumed that the \mathbb{M}_δ is changed. Since the current segment λ^k is now

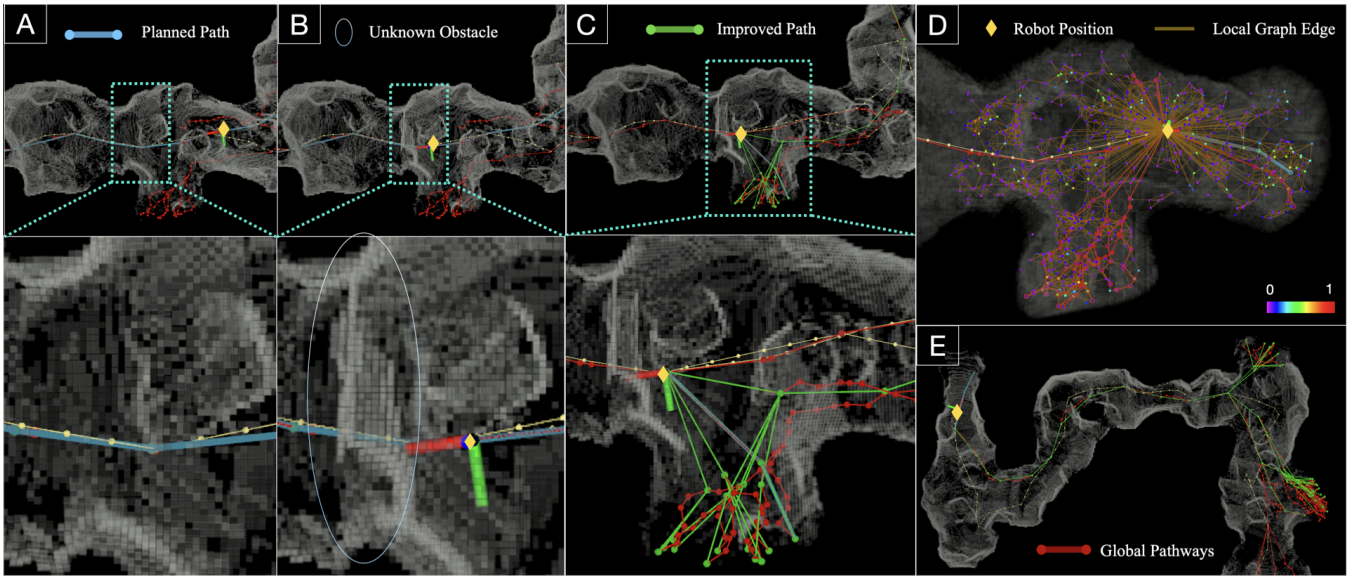


Fig. 4: STAGE planner’s adaptive re-planning behaviour in presence of unknown change to the global map. The obstacle in [B] is added when robot was out of line of sight. In [C] re-planned path ways are highlighted in green.

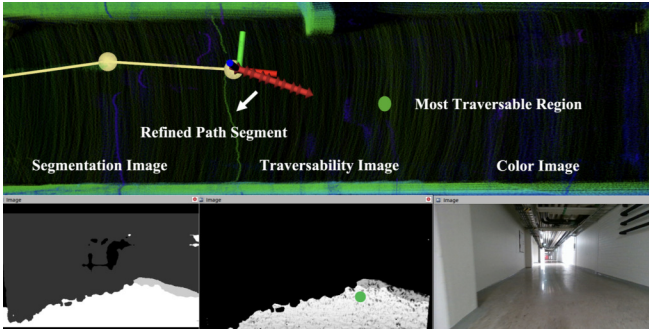


Fig. 5: Semantic Traversability ST image. The refined path segment (red arrow) is computed based on the most traversable zone depicted on the ST_{image} , where traversability is represented by a set of semantic labeled classes describing the smoothness of the terrain (e.g. flat, rough, obstacle, etc.).

untraversable, we sample new nodes in O_{ss} to find if an alternative path segment can be generated to plan the path around Υ_u . If no such segment is found, then the path way is marked untraversable as λ_{ut} in the P_G followed by global graph update \mathbb{G}_G that removes the edges that form λ_{ut} . The planner re-evaluates the updated global graph to compute new λ_G to continue exploration while registering the unknown change Υ_u in the global graph to further discard edges in newly discovered untraversable zone. A visual representation of this re-planning process is shown in the planning steps F, G and H of Fig. 3.

V. STAGE EVALUATION

The STAGE planner was evaluated in simulations and real life experiment to demonstrate A) Scalable efficient exploration in large SubT settings and B) Adaptive global re-positioning in the presence of unknown change in the explored map. The conducted simulations and experimental evaluation of stage planner can be watched at https://youtu.be/NC4r_hhfZ-M?si=cFwbdwqdlFaRnil_. STAGE planner is also platform agnostic, evaluated for both aerial robot in

simulation and legged robot in real life experiments. The simulations were carried out with in gazebo simulator using the RotorS [18] Robot Operating System (ROS) package. The robot is equipped with a Velodyne VLP16 3D LiDAR to generate point cloud as an input for STAGE planner. The goal of the simulation evaluation was to have the robot explore unknown subterranean cave environment that has multiple junctions and varying topology like narrow passages and large void-like structures. To demonstrate the adaptive global re-planning in presence of uncertainty in the explored map, we block the passage way through which the robot has planned global path. As shown in Fig. 4 [A] the robot has planned path in traversable space of the global map \mathbb{M}_G . As the robot executes the planned global path λ_G , we purposefully block the passage through which the robot has already planned λ_G . Normally in global re-positioning maneuver, the robots blindly follow the global or homing paths. As mentioned in section IV in contrast to the state-of-the-art approaches, the proposed planner segments the global path and evaluates each segment before execution to handle any uncertainty present. As shown in the Fig. 4 [B], the proposed method evaluates the immediate segment within point cloud visibility and discovers that a change Υ_u in the map has occurred. Then, the next step is to check if alternate paths can be found to reach the goal of λ_G . Since the passage way was completely blocked, the robot takes into consideration Υ_u and re-plans global path way to another global frontier in the vicinity. As highlighted in the Fig. 4 [C], the robot finds alternate improved pathways that account for newly discovered change Υ_u . As a result, the robot is able to safely globally re-position and continue exploration. A highlight of the information gain associated with frontier nodes of sub graph is shown in Fig. 4 [D]. The information gain scales from 0 to 1 and higher informative path in free space is planned as shown in Fig. 4 [D]. Finally in E we show instance of exploration where the robot is locally exploring

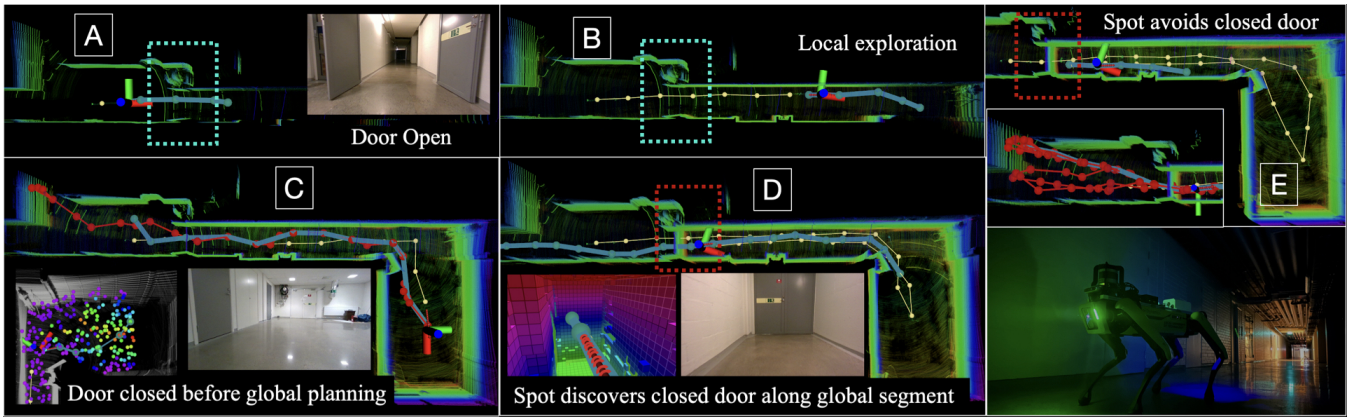


Fig. 6: Experimental evaluation of STAGE planner with Boston Dynamics spot robot.

and has candidate paths to global exploration are shown in green. The experimental evaluation of the STAGE planner was

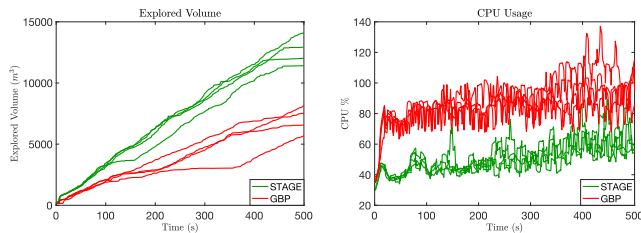


Fig. 7: Explored volume and Resource utilization comparison. performed in the underground corridors of university building, located at the Luleå University of Technology campus in Sweden. The hardware setup for the field tests used the Boston Dynamics Spot robot, configured with our own sensor and computation kit on top. The Spot kit also uses an Intel NUC 10 BXNUC10I5FNKPA2 for computation, a Velodyne Puck Lite 3d LiDAR, a RealSense D455 mounted forward and a VectorNav VN-100 IMU. To highlight the novelty of the STAGE planner we choose a scenario where the Spot robot locally explores a long corridor that is connected to open room at the end (replicating the void like topology) and while the robot is out of sight of the corridor, we purposefully block the passage by closing the door. In the Fig. 6 [A], when the Spot robot starts the exploration the door is open (area highlighted in blue box). As the mission progresses Fig. 6 [B] the Spot robot explores the corridor and reaches the end where open room is found. While the robot is out of line of sight as shown in Fig. 6 [C], the door is purposefully closed (area highlighted in blue box). At this instance, the Spot robot has planned a global re-positioning path to continue exploration in another corridor with the previous information that the passage is traversable. As the spot robot navigates along the global path λ_G , in Fig. 6 [D] it discovers an untraversable segment. No alternate paths could be found as the door was fully closed. Therefore, as shown in Fig. 6 [E] the Spot robot re-positions itself to explore residual frontier nodes in the previous corridor before completing the exploration mission. Since the STAGE planner is designed to be scalable and platform agnostic, we have compared the performance of in exploration and resource utilization with the State of the Art

graph based planner [GBP] [14]. We perform multiple runs of the STAGE planner and GBP to explore a subterranean cave environment with equal mapping and robot dynamics and control parameters. The results of the performance comparison are highlighted through Fig. 7. Since GBP is also a graph based planner and has global planning functionality, it explores considerable portion of the environment. However, due to STAGE planner’s direct point cloud visibility based rapid local exploration and efficient global re-positioning, STAGE planner shows comparatively higher exploration gain. Apart from the standard exploration volume gain metric, we also compare the CPU utilization of the two algorithms in the same runs. Since STAGE planner adapts a novel sub-graph overlap based global graph generation and re-planning, the STAGE planner consumes considerably less CPU to run both algorithm 1 and algorithm 2 in the same ROS node.

VI. CONCLUSION

Through this article we propose the STAGE planner, a scalable novel navigation scheme for large scale exploration in environments with dynamic scene changes. STAGE is a two layered graph based framework with memory efficient management of the local and global layers. More specifically, the local sub-graphs are adaptively tuned to local bounds based on the detected frontiers on the direct pointcloud visibility, enabling fast traversal towards local frontiers. The global graph is built using minimal node-edge information exchange only on the overlapping regions of sequential sub-graphs. STAGE is also traversability aware planer able to handle scene changes (e.g. blocked pathways) in previously explored areas. It adaptively updates the obstructed part of the global graph from traversable to not-traversable. Each path in the global graph is divided in multiple segments, where each one is collision checked through oriented sample space to remove edges from connected nodes in cases of obstructions. In such cases, the exploration behavior is directing the robot to follow another route in the global re-positioning phase through path-way updates in the global graph.

REFERENCES

- [1] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. Towards New*

- Computational Principles for Robotics and Automation*, pp. 146–151, IEEE, 1997.
- [2] B. Lindqvist, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, “Exploration-rrt: A multi-objective path planning and exploration framework for unknown and unstructured environments,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3429–3435, IEEE, 2021.
 - [3] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund, *et al.*, “Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge,” *arXiv preprint arXiv:2103.11470*, 2021.
 - [4] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, D. Heřt, M. Petrлік, T. Báča, V. Spurný, *et al.*, “Darpa subterranean challenge: Multi-robotic exploration of underground environments,” in *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29–31, 2019, Revised Selected Papers 6*, pp. 274–290, Springer, 2020.
 - [5] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascaričh, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, *et al.*, “Cerberus in the darpa subterranean challenge,” *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.
 - [6] N. Hudson, F. Talbot, M. Cox, J. Williams, T. Hines, A. Pitt, B. Wood, D. Frousheger, K. L. Surdo, T. Molnar, *et al.*, “Heterogeneous ground and air platforms, homogeneous sensing: Team csiro data61’s approach to the darpa subterranean challenge,” *arXiv preprint arXiv:2104.09053*, 2021.
 - [7] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, “Evaluating the efficiency of frontier-based exploration strategies,” in *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pp. 1–8, VDE, 2010.
 - [8] H. Umari and S. Mukhopadhyay, “Autonomous robotic exploration based on multiple rapidly-exploring randomized trees,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1396–1402, IEEE, 2017.
 - [9] A. Patel, B. Lindqvist, C. Kanellakis, A.-a. Agha-mohammadi, and G. Nikolakopoulos, “Ref: A rapid exploration framework for deploying autonomous mavs in unknown environments,” *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, p. 35, 2023.
 - [10] A. Patel, S. Karlsson, B. Lindqvist, C. Kanellakis, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, “Towards energy efficient autonomous exploration of mars lava tube with a martian coaxial quadrotor,” *Advances in Space Research*, vol. 71, no. 9, pp. 3837–3854, 2023.
 - [11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon” next-best-view” planner for 3d exploration,” in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1462–1468, IEEE, 2016.
 - [12] L. Lu, A. De Luca, L. Muratore, and N. G. Tsagarakis, “An optimal frontier enhanced “next best view” planner for autonomous exploration,” in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pp. 397–404, IEEE, 2022.
 - [13] T. Rouček, M. Pecka, P. Cizek, T. Petricek, J. Bayer, V. Šalansky, T. Azayev, D. Hert, M. Petrлік, T. Báča, *et al.*, “System for multi-robotic exploration of underground environments ctu-cras-norlab in the darpa subterranean challenge,” *arXiv preprint arXiv:2110.05911*, 2021.
 - [14] T. Dang, F. Mascaričh, S. Khattak, C. Papachristos, and K. Alexis, “Graph-based path planning for autonomous robotic exploration in subterranean environments,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3105–3112, IEEE, 2019.
 - [15] T. Dang, M. Tranzatto, S. Khattak, F. Mascaričh, K. Alexis, and M. Hutter, “Graph-based subterranean exploration path planning using aerial and legged robots,” *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, 2020.
 - [16] D. Duberg and P. Jensfelt, “Ufomap: An efficient probabilistic 3d mapping framework that embraces the unknown,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6411–6418, 2020.
 - [17] J.-C. Chen, “Dijkstra’s shortest path algorithm,” *Journal of formalized mathematics*, vol. 15, no. 9, pp. 237–247, 2003.
 - [18] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, “Rotors—a modular gazebo mav simulator framework,” *Robot Operating System (ROS) The Complete Reference (Volume 1)*, pp. 595–625, 2016.