

S-Nav: Semantic-Geometric Planning for Mobile Robots

Paul Kremer*, Hriday Bavle*, Jose Luis Sanchez-Lopez* and Holger Voos*[†]

* University of Luxembourg, SnT, 29 Av. J.F. Kennedy, L-1855 Luxembourg, Luxembourg

[†] University of Luxembourg, FSTM, 2 Avenue de l'Université, L-4365 Esch-sur-Alzette, Luxembourg

Abstract—Path planning is a basic capability of autonomous mobile robots. Former approaches in path planning exploit only the given geometric information without leveraging the inherent semantics within the environment. The recently presented *S-Graphs* construct optimizable 3D situational graphs incorporating geometric, semantic, and relational aspects between the elements to improve the overall scene understanding and the localization of the robot. But these works do not exploit the underlying semantic graph to improve the path planning for mobile robots. In this direction, we present *S-Nav* a novel semantic-geometric path planner for mobile robots. It leverages *S-Graphs* to enable fast and robust hierarchical high-level planning in complex indoor environments. The hierarchical architecture of *S-Nav* adds a novel semantic search on top of a traditional geometric planner as well as precise map reconstruction from *S-Graphs* to improve planning speed, robustness, and path quality. We demonstrate improved results of *S-Nav* in a synthetic environment.

I. INTRODUCTION

Mobile robots have gained a lot of traction in recent years and have seen widespread use in different industries such as construction, mining, etc., where they are used for autonomous inspection tasks. To date, they are mostly teleoperated or operated semi-autonomously under the supervision of a human operator. Fully autonomous operation could thus significantly reduce costs, however, several technical challenges such as perception, navigation, mapping, and localization are currently detrimental to this mode of operation. Mobile robots should not only create meaningful maps of the environment while localizing within it but also be able to exploit these maps to perform fast and efficient planning.

Traditionally, mobile robots build a geometric map [4], [5] of their environment using simultaneous localization and mapping techniques (SLAM) in combination with their on-board sensors [7] (e.g., LiDAR). Recently, we presented *S-Graphs* a novel graph-based semantic SLAM that combines traditional geometric SLAM with scene graphs [1], [2]. *S-Graphs* extracts the topological-relational information of the environment such as wall surfaces, rooms, and doorways including the topological connections between those semantic entities enabling the robot to reason about its environment in a way humans would, and showed promising results in terms of precise robot localization and high-level hierarchical map generation over a variety of datasets. However, this scene knowledge is not yet leveraged for performing more intelligent and faster path planning for mobile robots

To bridge this gap, we leverage the metric, semantic, and relational information in *S-Graphs* for the purpose of path

planning. We propose a novel hierarchical planner called *S-Nav* which leverages the semantic layer to improve planning on the geometric layer. First, we perform a semantic graph search utilizing the semantic elements within the *S-Graphs* to generate a sparse undirected graph of semantic elements such as rooms and doorways. The undirected global semantic graph is then divided into local subproblems which can be solved in parallel and pose a set of simpler problems to the underlying geometric planner. The main contributions of this work are:

- A novel hierarchical planner called *S-Nav* utilizing geometric, semantic, and relational information for faster planning.
- A semantic planner for faster global plans.
- A semantic subproblem solver further simplifies the global plan into local subproblems for the underlying geometric planner.

II. SYSTEM OVERVIEW

Figure 1 shows an overview of *S-Nav*, the proposed navigation system. The main blocks are the *Semantic Planner* (section III-A), the *Subproblem Solver* (section III-B), and the *Geometric Planner* (section III-D). The whole system builds on top of *S-Graphs*, summarized in section II-A.

A. Situational Graphs (*S-Graphs*)

S-Graphs is an optimizable graph structure built using online measurements such as LiDAR data or markers [1], [2], [6].

The graph structure consists of five layers that are summarized as:

Keyframe Layer: Composed by the robot's pose ${}^M\mathbf{x}_{R_i} \in SE(3)$ constrained by the robot's odometry measurements.

Wall Layer: Each room is composed of four planes extracted from onboard sensor measurements. They are constrained using pose-plane constraints.

Room Layer: A room is formed by its four planes constrained by a cost function consisting of the room center ${}^M\mathbf{y}_i \in \mathbb{R}^2$ and the distance between the opposite planar pairs.

Floor Layer: A floor is a collection of rooms optimized analogously to rooms by extracting the largest distance between the opposite planar pairs.

Doorway Layer: A doorway marks the physical, traversable connection between two rooms defined by a center point ${}^M\mathbf{d}_i \in \mathbb{R}^2$ and a width w_i , and is constrained by the mutual physical distance between the two rooms it connects.

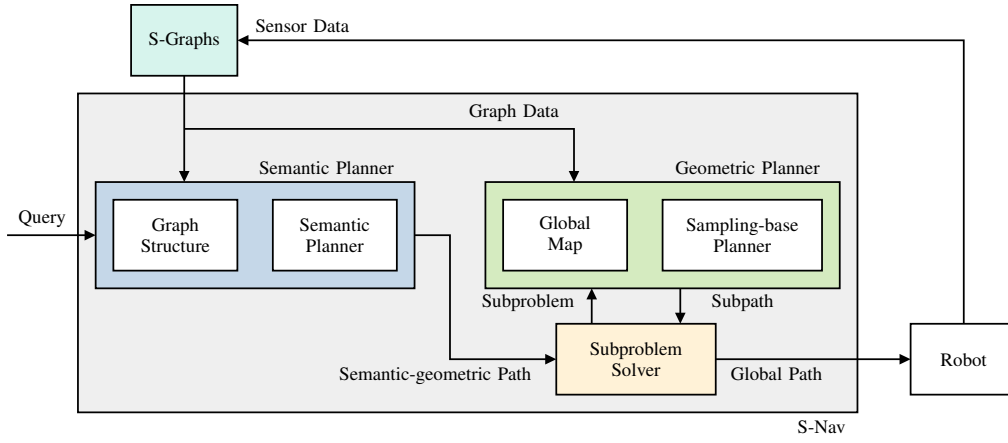


Fig. 1. Overview of *S-Nav*. A *query* (e.g., ‘go from current position to the kitchen’) is first handled by the *Semantic Planner* which provides an initial high-level semantic-geometric path based on the graph structure obtained from *S-Graphs*. Next, in the *Subproblem Solver*, the semantic-geometric path is subdivided into smaller, easier problems that are individually solved by the *Geometric Planner*. The individual paths between the subproblems are then reassembled into the final high-level path that is passed to the robot.

This work makes extensive use of the room, wall, and doorway layers. The presented architecture can, however, easily be expanded to include multiple floors and other semantic entities (e.g., objects).

III. S-NAV

S-Nav is our novel hierarchical semantic-geometric planning solution that combines *S-Graphs* with an informed geometric planner. This provides the following benefits over traditional, purely geometric planners:

- The geometric search can greatly profit from a rough initial guess provided by the semantic layer by virtue of constraining the areas the planner can visit and by providing subgoals toward the final goal.
- A query in natural form, e.g., ‘go from here to the kitchen’, can easily be mapped to a semantic-geometric problem.
- Handling forbidden areas such as closed doors or rooms that should not be traversed is trivial on the semantic layer, whereas it would require map changes on the geometric layer. Similarly, if a doorway is detected as untraversable, replanning is very fast as the doorways node can easily be removed from the graph.

The structure of *S-Nav* is depicted in fig. 1. Its main parts are formed by the *Semantic Planner* that cascades into the *Geometric Planner* via the *Subproblem Solver* (SPS). The final path is then passed to the robot, potentially via additional layers such as local planning, trajectory generation, and motion control.

A. Semantic Search

The scene graph structure of *S-Graphs* encodes a high-level representation of the environment the robot is operating in. Herein, this scene graph is converted into an undirected graph connecting the semantic elements of the scene. The

connections (edges) between the elements have an associated cost, i.e., for the room-to-doorway connections, a cost

$$c_{dr} = \|{}^M\mathbf{y}_i - {}^M\mathbf{d}_i\|^2 + p_d \quad (1)$$

is assigned, consisting of the distance between the center point of the room and the associated doorway, plus a fixed penalty for doorway crossing. The fixed penalty p_d can be used to prefer a slightly longer path with fewer (potentially closed) doorways.

Generally, the graph is sparse, featuring only a small number of nodes and edges. As such, a shortest-path search using, e.g., A* is virtually free compared to a full search on the geometric layer.

To a given *query* ‘ $\mathbf{p}_s \rightarrow \mathbf{p}_g$ ’, the semantic planner provides a *solution* of type

$$\mathbf{p}_s \rightarrow \mathbf{d}_k \rightarrow \mathbf{d}_{k+1} \rightarrow \dots \rightarrow \mathbf{d}_{k+n} \rightarrow \mathbf{p}_g, \quad \text{via: } F_R = R_s \cup R_i \cup \dots \cup R_g, \quad F_R \subset \mathbb{R}^3, \quad (2)$$

where $\mathbf{p}_s, \mathbf{p}_g, R_s, R_g$ are the start and goal positions and rooms, $\mathbf{d}_{(\cdot)}$ the doorways to traverse along the route, and $R_{(\cdot)}$ are the rooms (free space) the robot has to pass through to reach its destination. F_R is the newly constructed restricted state space passed to the geometric sampler. An example of such a search is visualized in fig. 2.

B. Semantic Search with Subproblems

The global problem in eq. (2) can further be simplified into a set of local subproblems that are solved in parallel and, individually, pose a simpler problem to the geometric planner:

$$\begin{aligned} 1 : \mathbf{p}_s &\rightarrow \mathbf{d}_k, & \text{via: } R_s \\ 2 : \mathbf{d}_k &\rightarrow \mathbf{d}_{k+1}, & \text{via: } R_{i+1} \\ 3 : \mathbf{d}_{k+1} &\rightarrow \mathbf{d}_{k+2}, & \text{via: } R_{i+2} \\ &\dots & \\ n : \mathbf{d}_{k+n} &\rightarrow \mathbf{p}_g, & \text{via: } R_g \end{aligned} \quad (3)$$

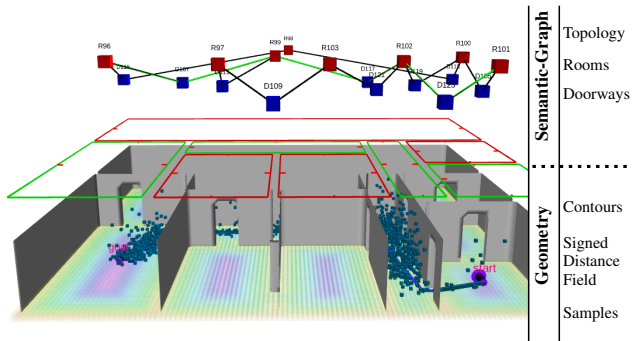


Fig. 2. The semantic-graph layer is an undirected topological graph of rooms and doorways extracted from features segmented by *S-Graphs*. The result of this first layer search cascades into the geometric layer which is formed by OMPL and Voxblox. The contour layer formed from the extracted walls reduces the valid state space of the geometric planner to do informed decisions on where to sample. The semantic path is highlighted in green together with the room contours that have to be traversed along the path.

Therefore, akin to informed geometric planners (e.g., informed RRT* [3]), herein the semantic planner adds an additional layer of information that the geometric planner can profit from. The subproblems are solved by the *Subproblem Solver* in conjunction with the geometric planner. The resulting individual path segments are joined to a final, global path. If the resulting path requires updating, e.g., due to a blocked path, the *Subproblem Solver* can efficiently reevaluate the changed or newly created subproblems.

C. Global Map

Instead of relying on raw sensor readings, *S-Nav* features a global map reconstruction module that builds an accurate global map from *S-Graphs* data, which in turn is generated on the fly by the robot, resp. provided to the robot if the environment is already fully mapped. The global map is kept relatively simple, i.e., not featuring obstacles as this problem is more effectively handled by the reactive planner on a local map.

S-Graphs provides the planes associated with each room, including the doorways that mark the connection between two rooms. In the first step, for each room, the vertical planes (walls) are converted to a closed 2D contour which encompasses the free space within a room. Room contours serve two purposes: *First*, to restrict the geometric planner's sampler to sample only in areas that effectively contribute to the final path. *Second*, to build an optimistic, clutter-free (yet accurate), signed distance field representation of the physical environment that forms the basis for the geometric planner. Similarly, doorways are added between the two closest walls of the two associated rooms.

D. Geometric Search

The geometric search within *S-Nav* features state-of-the-art geometric planners provided by the *Open Motion Planning Library* (OMPL). Within *S-Nav* we use sampling-based planners (e.g., PRM, RRT), that create random (sometimes with a heuristic) samples within the valid bounds of the state space. A

priori, for any given problem, the whole global map has to be considered. Therefore, a large number of samples is required to converge toward the optimal path. Constraining the sampler to sample within the rooms that have to be visited along the semantic path, greatly enhances the convergence rate of the planner as no samples are wastefully created in areas that are of no interest. Furthermore, using the SPS that decomposes the global problem into a set of local problems effectively exploits the rapid convergence of certain planning algorithms (e.g., IRRT*) for the resulting simpler type of problems. The problem is illustrated in fig. 3.

IV. EVALUATION

A. Methodology

A synthetic map (17m × 15m) with 8 rooms and 10 doorways was created and passed to the recently presented *iS-Graphs* [6], an *S-Graphs* extension that supports architectural (BIM) data. Within this environment, the three strategies (IRRT*, IRRT*+*S-Graphs*, and *S-Nav*: IRRT*+*S-Graphs*+SPS) shown in fig. 3 were benchmarked by performing 1000 queries for each. As OMPL termination criteria, a timeout of 0.1s was specified. The timeout is equally divided over all subproblems for the test series involving the SPS. Recorded were the number of samples created within the allocated time as well as the final path length. The measurements were performed on a workstation equipped with an Intel Core i9-11950H.

B. Results and Discussion

The results are given in fig. 4. It is clear that IRRT* alone delivers the most inconsistent results with the widest spread. On average, it also had the least number of samples generated. Restricting the sampled regions with the *S-Graphs* knowledge, significantly improved the consistency of the results. Further, using *S-Graphs* and the SPS in combination with IRRT* yielded consistently the shortest path, and was also able to generate significantly more samples. The higher number of samples is caused by the comparatively cheaper state and motion validity checks based on the contours rather than the signed distance field alone.

V. CONCLUSION

Leveraging the geometric-semantic knowledge contained in *S-Graphs* for planning can greatly enhance the performance of the underlying geometric planner. Herein, we presented *S-Nav*, a novel semantic-geometric planner that features a hierarchical planner architecture that showed to significantly improve planning speed, resp., the consistency of the generated paths within a given time frame. Furthermore, we showed that decomposing the global problem into a set of local problems can be used to effectively leverage the rapid convergence of (informed) sampling-based planners.

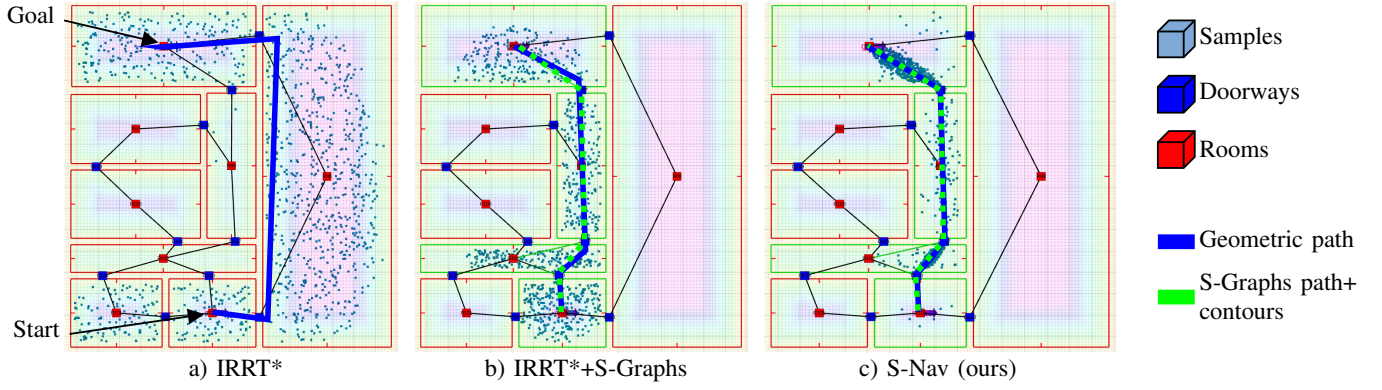


Fig. 3. a) The naïve approach using IRRT* creates an excessive amount of samples and yet ends up with a suboptimal path. b) Restricting IRRT* to sample within the rooms that are part of the optimal path greatly enhances the solution with fewer samples. c) Restricting IRRT* to sample within the rooms parts of the solution and decomposing the overall problem in subproblems yields the best result (least amount of samples) as it effectively exploits the rapid convergence of IRRT* for the resulting simpler problems.

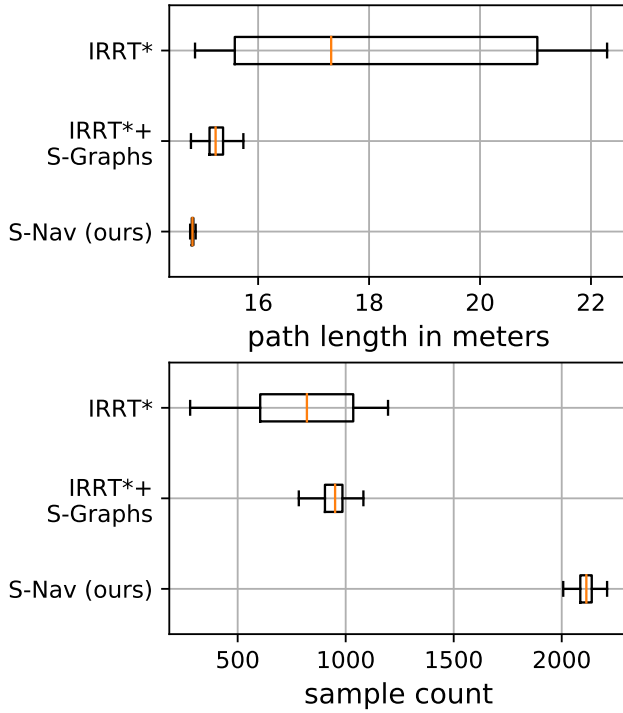


Fig. 4. Results: The semantic planner greatly improves the consistency of the results by reliably finding the shorter path compared to plain IRRT*. The *S-Nav* strategy involving IRRT*, *S-Graphs*, and the SPS created significantly more samples in the allocated time and consistently found the shortest path.

REFERENCES

- [1] Hriday Bavle, Jose Luis Sanchez-Lopez, Muhammad Shaheer, Javier Civera, and Holger Voos. Situational Graphs for Robot Navigation in Structured Indoor Environments. *IEEE Robotics and Automation Letters*, 7 (4):9107–9114, oct 2022. ISSN 2377-3766. doi: 10.1109/LRA.2022.3189785. URL <https://ieeexplore.ieee.org/document/9826367/>.
- [2] Hriday Bavle, Jose Luis Sanchez-Lopez, Muhammad Shaheer, Javier Civera, and Holger Voos. S-Graphs+: Real-time Localization and Mapping leveraging Hierarchical Representations. (1):1–10, dec 2022. URL <http://arxiv.org/abs/2212.11770>.
- [3] Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2997–3004. IEEE, sep 2014. ISBN 978-1-4799-6934-0. doi: 10.1109/IROS.2014.6942976. URL <https://ieeexplore.ieee.org/document/6942976/>.
- [4] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2017-Sept, pages 1366–1373. IEEE, sep 2017. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8202315. URL <http://ieeexplore.ieee.org/document/8202315/>.
- [5] Helen Oleynikova, Zachary Taylor, Roland Siegwart, and Juan Nieto. Sparse 3d topological graphs for micro-aerial vehicle planning, 2018.
- [6] Muhammad Shaheer, Jose Andres Millan-Romera, Hriday Bavle, Jose Luis Sanchez-Lopez, Javier Civera, and Holger Voos. Graph-based Global Robot Localization Informing Situational Graphs with Architectural Graphs. mar 2023. URL <http://arxiv.org/abs/2303.02076>.
- [7] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41 (2):401–416, feb 2017. ISSN 0929-5593. doi: 10.1007/s10514-016-9548-2. URL <http://link.springer.com/10.1007/s10514-016-9548-2>.