Topology-based Visual Active Room Segmentation

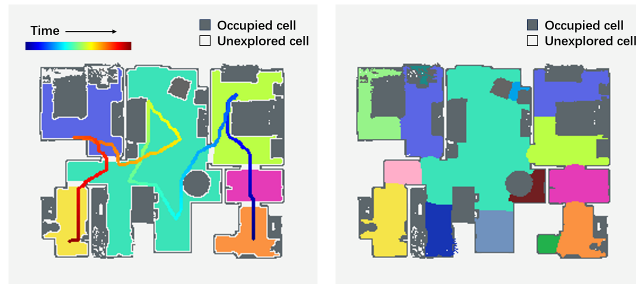
Chenyu Bao, Junjie Hu, *Member, IEEE,* Qiu Zheng and Tin Lun Lam, *Senior Member, IEEE*

***Abstract*— Room segmentation plays a significant role in scene understanding, semantic mapping, and scene coverage for robots navigating in real-world indoor environments. However, most previous works take a passive segmentation that requires a complete and uncluttered grid map as input, often resulting in lower segmentation accuracy and cannot be deployed in unknown environments. In this paper, we propose an active room segmentation framework that can enable a robot to incrementally and autonomously perform room segmentation in cluttered indoor environments. Our framework consists of three key components: i) a door extraction module where a visual semantic feature, specifically, door, is extracted to better identify rooms in cluttered environments, ii) a within-room exploration module that detects frontiers within the currently exploring room, and iii) a topological module that represents connectivity between rooms and determines next room for exploration. We show through experiments that the proposed method depicts two distinct advantages against existing methods in segmentation accuracy and autonomy. The code is available at `https://github.com/FreeformRobotics/Active_room_segmentation`.**

I. Introduction

For robots working indoors, occupancy grid map is one of the most widely used representations of the environment. It discretizes the continuous environment into plenty of square cells (cubical volumes in 3D case) of the same size. Each cell stores a probability indicating how likely the cell is free, occupied, or unknown. The occupancy grid map is quite straightforward for human users but not for robots. Thus, a further process is required to make it easier for robots to utilize in downstream tasks like human-robot interaction or cleaning [1], [2]. In indoor environments, this process can be specified as room segmentation.

Room segmentation serves as a critical process that furnishes higher-level information by dividing the occupancy grid map into semantically meaningful segments. In previous research, this segmentation can be broadly categorized into: offline and incremental, both typically reliant on *a prior* knowledge regarding the indoor building structure. Offline segmentation techniques, for instance, are predicated on the identification of specific architectural features, like narrow passages [3], [4] or potential door locations [2], [5], to divide contiguous regions into distinct rooms. In contrast, incremental methods, as exemplified by DUDE [6] and the work of Choi et al. [7], assume that rooms extracted from the maps exhibit convex shapes. They propose incremental methodologies that assess the convexity of newly scanned areas to determine if further decomposition is warranted.



(a) Ours (b) Voronoi graph-based method

Fig. 1: Comparison between our active method and the widely used Voronoi graph-based offline segmentation method [3].

Existing room segmentation methods exhibit two primary limitations. Firstly, they often disregard valuable visual cues and rely solely on occupancy maps as input. These maps offer only limited geometric information, such as convexity or size. These constraints prove insufficient when confronted with cluttered indoor environments, where substantial furniture pieces like tables, sofas, or beds can mislead algorithms into fragmenting a single room into multiple segments. Secondly, prior methodologies lack an autonomous exploration strategy. Offline methods assume the availability of high-quality, complete occupancy grid maps as input. Conversely, most incremental approaches necessitate human intervention to update these occupancy grid maps. The former issue adversely impacts room segmentation accuracy, while the latter impedes the autonomy of robotic systems. Despite the feasibility of simply concatenating a room segmentation module to an active exploration module directly for autonomy, it leads to inefficiency as each module works independently towards its own goal. The exploration tends to prioritize larger unexplored areas without thoroughly exploring the current room, impeding the room segmentation from swiftly getting the final result. Additionally, most previous off-the-shelf room segmentation methods typically return a segmented grid map, which is challenging to effectively utilize within the context of the exploration task.

This paper presents an active room segmentation framework integrating both exploration and segmentation in a

C.Bao, J.Hu, Q.Zheng and T.L.Lam are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China.

J.Hu and T.L.Lam are with Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), The Chinese University of Hong Kong, Shenzhen, China.

C.Bao and J.Hu contribute equally.

T.L.Lam is the corresponding author. `tllam@cuhk.edu.cn`

This work was partially supported by the Guangdong Natural Science Fund under Grant 2024A1515010252, Shenzhen Science and Technology Program under Grants JCYJ20220818103000001 and JCYJ20241202124010014, and the Shenzhen Institute of Artificial Intelligence and Robotics for Society under Grants AC01202101103.

complementary way. It ensures a room-by-room exploration strategy for better segmentation results while the room segmentation results are generated in the form of a topology map to facilitate the exploration process. To maintain segmentation performance and construct grid maps in cluttered environments, the framework takes the RGBD images as its input and returns a topology map and action instructions for active exploration. Our approach can be distilled into three key modules: i) a two-stage door extraction module that can identify doors from cluttered indoor environments, ii) a within-room exploration module that enables the robot to fully explore a single room, and iii) a topological module that updates a topology map based on the detected door locations and tells the robot the next room to visit once it completes its exploration of the current one. Through a repetitive cycle of these modules, our method autonomously conducts online room-by-room segmentation, enabling maximal exploration of unfamiliar environments.

To validate the effectiveness of our method, we conducted extensive experiments using the real-world Gibson dataset [8]. Our results demonstrate that our method consistently outperforms the previous state-of-the-art approach, exhibiting a 3.9% improvement in recall and a substantial 10.2% increase in precision for segmentation, while striking a balance between exploration completeness and efficiency. A visual comparison between our method and the widely used Voronoi graph-based method for offline segmentation [3] is illustrated in Fig. 1. As seen, our method yields distinct advantages in segmentation accuracy and autonomy.

In summary, our contributions include:

- We propose a novel active room segmentation framework that undertakes room segmentation and exploration in a complementary way.
- The framework efficiently reuses the RGBD images required by the SLAM module to maintain a robust segmentation performance under cluttered environments.
- We introduce a topology map construction module to represent the segmentation results, modeling the connectivity relations between rooms to facilitate the exploration.

II. Related work

A. Offline Segmentation

Offline segmentation methods typically operate on complete scene maps and extract features from these maps for segmentation. For example, regions of narrow passages are one of the most intuitive and popular features. Thrun et al. [9] first attempted to utilize the Voronoi graph to locate the narrow passages, referred to as critical points in their work. Later, Voronoi graph-based methods [3] have become the predominant approach for generating floor plans. Other technologies like morphology operation [1], [4] or distance transformation [3], [10], [11] have also been used for locating narrow passages in the scene. The virtual door is a more specific instance for narrow passages. It was first proposed by Myung et al. [2], indicating possible positions of doors derived from the grid map. This concept is also used in [5]. Luperto et al. [12] proposed to extract the main directions from cluttered scenes by Discrete Fourier Transform and use heuristics to do segmentation on each direction. Besides finding those semantic meaningful features, other researchers also resort to spectral clustering [13], [14], [15] or deep learning methods [16], [17] to utilize more implicit features for segmenting the map.

However, offline methods are limited by their need for a complete map as input, making them unsuitable for deployment on robots navigating unknown environments. Furthermore, these methods often struggle to effectively incorporate visual cues, such as RGB images, which results in lower segmentation accuracy, particularly in complex real-world scenarios. For instance, virtual door-based methods rely solely on geometric cues extracted from input grid maps to identify doors. In contrast, our approach leverages both visual (RGB) and geometric (depth) cues for door detection, enhancing its performance.

B. Incremental Segmentation

Incremental segmentation methods, on the other hand, rely on information obtained from new areas. Buschka et al. [18] designed a morphology-based room detection sensor that exclusively segments the newly scanned area and discards the older information. Another viable approach for incremental segmentation is through convexity [6], [7], which assumes that rooms in the scene are all convex in shape. Learning methods [19], [20] are also highly suitable for incremental segmentation, as the network can efficiently classify grid cells in real-time using sensor inputs like Lidar scans or RGB images. However, all of the aforementioned methods require human intervention to update the occupancy grid map. Although a few studies [21], [22] have attempted to integrate the segmentation task with the exploration task, they still rely on simple geometric features extracted from the occupancy maps, resulting in suboptimal accuracy.

The most related work to ours is [23], which is the first and only existing method enabling autonomous robot exploration and scene segmentation with door detection. However, our approach diverges in two crucial aspects. Firstly, while [23] uses a basic Hough line detector on depth images for door detection, we employ a deep learning-based door detection module working on RGBD inputs, enhancing accuracy. Secondly, our method excels at ensuring completeness during room exploration while [23] often fails in confined spaces. Consequently, [23] only demonstrated its effectiveness in simple and textureless scenarios. On the contrary, the proposed approach can yield superior performance in more complex scenarios.

C. Active Exploration

Active exploration enables the robots to gather required information (e.g., map) in unknown environments autonomously. The concept of “frontier” [24] delineates the boundary between known and unknown areas and serves as a widely adopted Region of Interest (RoI) for active exploration strategies. The Wavefront Frontier Detector (WFD)

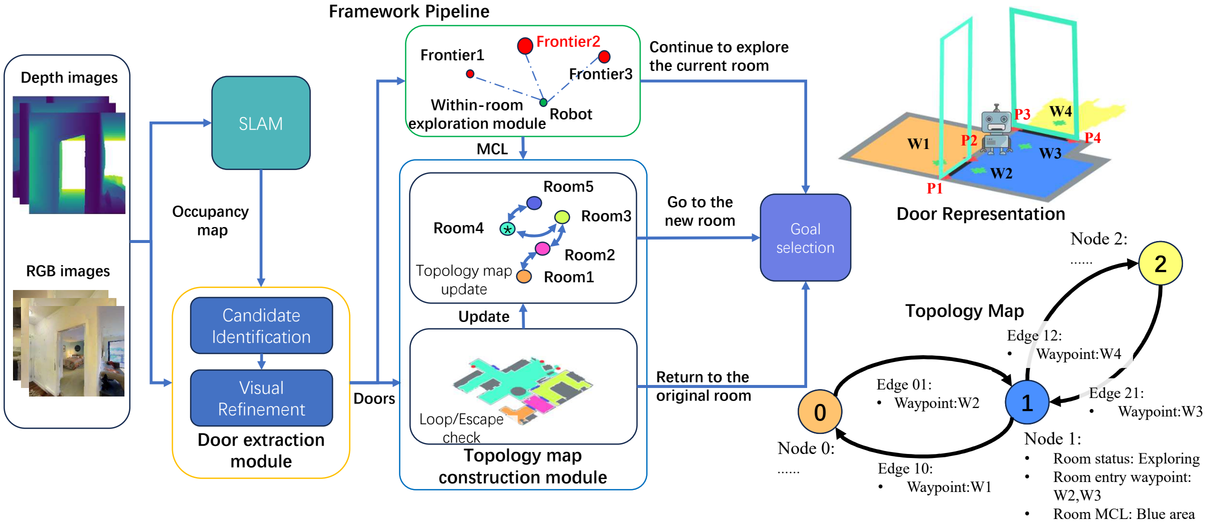


Fig. 2: Overview of the proposed framework, including illustration of door representation on the top-right and topology map on the bottom-right. To represent the door, a pair of points, e.g., P_1 and P_2, is used to represent a door, and similarly, two points, e.g., W_1 and W_2, are used to represent waypoints for passing through the door. In the topology map, different attributes are stored in each node and edge to help the robot better understand the scene.

[25] identifies frontiers through the implementation of a breadth-first search (BFS) across all unoccupied cells. However, it's worth noting that the processing time of WFD tends to increase as the map expands. To mitigate this, the Fast Frontier Detector (FFD) [25] expedites the process by exclusively searching within the most recent laser scans. In a similar vein, Sun et al. [26] restrict their frontier detection to specific sub-maps. Another innovative approach presented in [27] introduces a novel detection method based on the random sampling of sparse frontier points using Rapidly-exploring Randomized Trees (RRT) [28]. In recent years, there has been a growing interest in utilizing reinforcement learning techniques to identify RoIs [29], [30], [31]. However, learning-based methods require significant scenarios for training to ensure good generalizability and currently, it is still difficult for real-world deployment.

III. Methodology

This section presents details of the proposed active room segmentation framework in Fig. 2.

A. Door Extraction Module

For door extraction, we identify two points representing the location of each door (red points in Fig. 2). As door extraction in cluttered scenarios is difficult, we propose a two-stage method to extract doors accurately. We generate potential door candidates in the first stage and refine them in the next stage through a visual door detection network.

1) Identifying Door Candidates from Occupancy Map: We first adopt a ray-casting method inspired by [32] to identify N potential points $\mathcal{P}_{\mathbf{1st}} = \{P_j\}_{j=1}^{N}$ that are likely to be points representing door locations. This method casts rays that start from the robot's current location on the occupancy map and stop when intersecting with obstacles or reaching the length limit. Candidates can be easily detected by comparing gaps between lengths of adjacent rays.

2) Refinement with Visual Door Detection: The ray-casting method usually leads to many false results, as features from occupancy grid maps are insufficient to distinguish gaps caused by doors or large obstacles. Therefore, in the second stage, we leverage a deep neural network (DNN) to predict doors' bounding boxes from RGB images to filter out those false results. It can be represented as:

$$\mathbf{b_i} = \mathcal{F}\left(\mathbf{x_i}; \theta\right), \tag{1}$$

where $\mathbf{x_i}$ denotes a RGB image, $\mathcal{F}$ is the trained model with fixed parameters θ for door detection.

The above door detection is conducted repeatedly from 12 different views across 360 degrees once a robot reaches its target point during exploration. To align the bounding boxes with the door candidates, we project the bounding boxes into the occupancy map to generate points of door locations as follows:

$$\mathcal{P}_{\mathbf{2nd}} = \bigcup_{i=1}^{i=12} \psi(\mathbf{m_i} \otimes \mathbf{y_i}), \tag{2}$$

where $\mathbf{m_i}$ denotes a binarized bounding box and $\mathbf{y_i}$ is the corresponding depth map. $\otimes$ and ψ denote the element-wise multiplication and the operation for projection to the occupancy map, respectively. Then, more accurate door locations can be identified by computing the joint set of $\mathcal{P}_{\mathbf{1st}}$ and $\mathcal{P}_{\mathbf{2nd}}$. After visual refinement, we extract pairs of location points according to the distance and surrounding obstacles. To complete the door representation shown in Fig. 2, waypoints are obtained by choosing two near and free cells on each side of the door.

B. Within-room Exploration Module

To confine the searching area within the currently exploring room, the Wavefront Frontier Detector (WFD) [25] is slightly modified to make the iterative searching process stop at occupied cells or cells outside the current room. As a result, all free cells searched during this process are stored in the Map Close List (MCL), forming the explored area of the current room. For selecting the most informative frontier, we cluster all detected frontiers into groups by DBSCAN (Density-Based Spatial Clustering of Applications with Noise) after the searching process is done and assess each group based on their size and distance to the robot's current position using the cost function given in [33]. The above process will be repeated until no free cells are left in the exploring room.

C. Topology Map Construction Module

To aid in exploration, we construct and continuously update a topology map, as depicted in Figure 2. This map comprises individual nodes that correspond to distinct rooms in the scene, each possessing three crucial attributes: i) *Room Status*: Take on one of three exclusive states: “Explored”, “Exploring”, or “Unexplored”. ii) *Room Entry Waypoint*: Help the robot locate the entry point of the respective room, aiding navigation. iii) *Room MCL*: Indicate the extent of exploration within a room, offering insights into the explored area's coverage. Additionally, adjacent nodes in the topology map are linked by oppositely directed edges, representing connecting doors between rooms. Each edge includes a single waypoint to guide the robot to its destination node.

1) Loop/Escape Check: When the robot detects a door either after passing through it (referred to as an “escape”) or when there exists more than one door connecting two rooms (referred to as a “loop”), the simple connection of an 'Unexplored' node to the 'Exploring' node can yield incorrect outcomes. To avoid this, it's crucial to verify for loops or escapes before updating the map with the help of MCL information provided by the within-room exploration module. If the entry points to rooms linked to the 'Exploring' node cannot be found in the MCL, it means the robot is in an escape scenario. Conversely, if entry points of other nodes can be found in the MCL, it indicates a loop situation.

2) Topology Map Update: When a new door becomes perceptible, the conventional procedure entails the creation of a novel “Unexplored” node, establishing connections to the “Exploring” node via two oppositely directed edges. In this process, the two waypoints associated with the door are allocated distinctively to the freshly introduced node, the “Exploring” node, and the corresponding edges as Fig. 2 shows.

However, when the robot confronts an escape situation, a reversal in waypoint assignment is necessitated. If the robot also identifies other doors, then these nodes are linked to the “Unexplored” node to which the robot is escaping, as opposed to the “Exploring” node. When the robot encounters a loop scenario, nodes whose room entry points are included by the MCL of the “Exploring” node undergo removal from the topology map. Subsequently, the attributes and edges belonging to these eliminated nodes are seamlessly integrated into the “Exploring” node, facilitating a coherent representation of the map's structure.

D. Overall Active Exploration Policy

Utilizing the meticulously constructed topology map as a critical resource, the robot adeptly determines its forthcoming objectives, guided by the following set of principles:

- Escape situation: When escaping the current room, the robot's next destination is determined by the nearest waypoint affiliated with the original node.
- Keep exploring: In cases where the current room remains incompletely explored, the robot designates a ”frontier” within the room as its next objective.
- Head for the next room: When the robot ascertains that the current room has been entirely explored, it strategically designates the waypoint of the nearest 'Unexplored' node as its subsequent goal.
- Completion of exploration: The termination of the active room segmentation process occurs when all nodes have been categorized as 'Explored', signifying comprehensive scene exploration.

The distance between the robot and the node is estimated by calculating a path from the 'Exploring' node to the 'Unexplored' node, followed by summing the distances between adjacent waypoints stored within the connecting edges along the selected path.

IV. Experiment

A. Implementation Details

For quantitative comparisons, we follow existing methods [34], [35] that conduct experiments on the real-world Gibson dataset [8] driven by the Habitat Simulator [36]. The robot in Habitat is equipped with an odometry sensor and an RGBD camera whose detection range is configured to 3 meters. To drive the robot towards goal points, the Fast Marching Method [37] will first calculate the shortest path towards goal points, and action is then generated in the same way as [38]. The robot's movement is governed by three fundamental actions: i)move forward by 25cm, ii) turn left by 30 degrees, and iii) turn right by 30 degrees. For map construction, the SLAM module employs the map builder[1] provided by ANS [39]. For the door extraction module, a transformer-based door detection network [40] is utilized to predict bounding boxes of doors from RGB images.

B. Segmentation Performance

1) Test Scenes: To quantify the performance of room segmentation, ten complex real-world scenes, as shown in Table. I, are chosen from the Gibson dataset. We manually annotate each room to generate ground truths for quantitative comparisons.

2) Baselines: We compare our method with all existing representative offline and incremental approaches. Among the offline methods, we include: ROSE[2] [12], the Morphological method [3], Distance method [3], Voronoi method [3], and the offline DUDE [6]. For incremental methods, we compare with the incremental DUDE [6] and Gomez

[1]https://github.com/devendrachaplot/Neural-SLAM/blob/master/env/utils/map_builder.py

TABLE I

AVERAGE RECALL AND PRECISION WITH RESPECTIVE STANDARD DEVIATION OF EACH SCENE

		Our method (Incremental)	DUDE[6] (Incremental)	Gomez et al.[23] (Incremental)	DUDE[6] (Offline)	ROSE[12] (Offline)	Morph[3] (Offline)	Distance[3] (Offline)	Voronoi[3] (Offline)
Cantwell(N=7)	R(%)	**93.6(1.1)**	61.2(4.5)	60.1(19.7)	61.5(2.2)	63.7(8.4)	80.6(2.2)	78.9(6.1)	58.1(6.8)
	P(%)	**96.3(1.6)**	85.6(5.3)	62.9(10.2)	84.5(4.9)	66.8(6.6)	80.6(3.2)	79.0(3.3)	71.8(5.6)
Swromville(N=6)	R(%)	**92.9(6.7)**	62.7(9.5)	72.4(28.0)	62.7(11.4)	44.2(3.8)	82.0(6.9)	84.2(0.7)	43.0(3.9)
	P(%)	**94.3(3.7)**	80.5(4.3)	66.9(8.5)	80.5(1.7)	60.4(4.1)	73.9(1.2)	67.8(1.0)	73.9(3.1)
Scioto(N=8)	R(%)	90.0(4.3)	73.7(2.4)	69.0(13.8)	75.7(3.8)	76.3(5.3)	93.5(0.8)	**94.8(1.5)**	80.7(4.2)
	P(%)	**95.6(5.4)**	81.0(8.4)	37.9(5.5)	81.3(5.3)	75.8(3.8)	82.9(3.8)	83.1(3.2)	81.2(3.0)
Eastville(N=6)	R(%)	**94.7(2.1)**	64.7(0.6)	89.3(8.9)	61.1(2.3)	62.2(9.1)	72.7(0.7)	89.6(0.1)	66.4(8.6)
	P(%)	**81.0(2.5)**	76.3(1.7)	62.0(8.5)	77.0(3.8)	67.9(5.4)	76.8(1.6)	64.1(6.1)	77.0(2.5)
Dunmor(N=6)	R(%)	99.0(0.7)	74.9(5.3)	53.4(27.5)	75.6(1.0)	71.4(9.7)	83.9(0.2)	**99.3(0.6)**	79.7(5.0)
	P(%)	**92.2(9.2)**	87.4(5.5)	66.9(11.9)	84.2(1.8)	70.4(4.6)	75.6(0.5)	81.3(0.6)	78.7(1.6)
Colebrook(N=6)	R(%)	97.1(0.8)	89.5(0.6)	88.3(10.2)	92.5(0.9)	89.3(3.8)	94.1(0.6)	**99.2(0.4)**	93.7(1.0)
	P(%)	**95.9(3.2)**	87.7(2.8)	69.2(9.5)	90.1(0.8)	78.9(2.2)	84.3(2.1)	86.7(0.1)	89.4(1.2)
Nicut(N=6)	R(%)	95.9(3.2)	72.3(4.9)	91.8(3.4)	73.9(3.8)	56.7(6.1)	95.8(1.1)	**98.6(0.7)**	65.6(7.0)
	P(%)	**98.9(0.5)**	85.2(2.9)	61.9(9.3)	91.0(0.4)	73.1(3.3)	83.2(0.8)	84.2(2.6)	76.4(4.0)
Quantico(N=6)	R(%)	**96.7(1.0)**	76.9(1.4)	81.8(13.0)	78.6(1.1)	59.1(5.5)	85.5(2.4)	89.5(0.5)	52.6(4.7)
	P(%)	**94.5(3.5)**	77.7(3.3)	58.2(12.5)	78.1(3.1)	69.7(4.1)	81.8(1.5)	79.9(0.8)	77.7(3.2)
Oyens(N=5)	R(%)	**98.5(0.8)**	58.4(9.8)	86.8(5.4)	56.4(1.4)	77.1(10.9)	76.9(0.2)	90.6(3.4)	62.6(3.6)
	P(%)	**98.2(1.0)**	92.7(2.2)	76.3(12.6)	91.8(0.8)	73.6(4.8)	70.0(0.5)	79.8(3.7)	81.8(1.5)
Hambleton(N=5)	R(%)	**98.9(0.8)**	78.3(0.8)	93.8(2.8)	80.7(0.2)	59.5(11.6)	86.5(0.4)	90.7(0.4)	49.8(5.5)
	P(%)	**98.7(1.0)**	90.0(0.6)	75.7(3.4)	87.6(0.3)	70.1(2.5)	70.5(0.4)	76.1(0.5)	78.2(1.6)
Average	R(%)	**95.4(4.0)**	71.3(10.1)	77.6(20.1)	71.9(11.2)	66.2(14.1)	85.5(7.7)	91.5(6.9)	65.9(15.8)
	P(%)	**94.5(6.2)**	84.1(6.5)	62.5(14.1)	84.3(5.8)	70.7(6.4)	78.4(5.3)	78.4(7.3)	78.5(5.5)

et al.'s method [23] we reproduced on Habitat [36] due to source code unavailability. For the remaining methods, we conduct validation using the source code provided by their respective papers. Considering that most of these prior methods necessitate complete maps as inputs, we preserved the maps generated by our approach and employed them as inputs for these methods. Additionally, we ensured that within each room of the scene, we selected a starting point that guaranteed coverage rates exceeding 99%.

3) Results: We employ precision and recall as the evaluation metrics for room segmentation. Quantitative comparison results can be found in Table. I, where N represents the number of rooms in each scene and the best and second-best are marked in bold and blue respectively. Our method significantly outperforms other approaches by achieving better performance in six scenes out of ten in both recall and precision. We achieved 95.4% in recall and 94.5% in precision on average, leading existing approaches by 10.2% to 32% in precision and 3.9% to 29.5% in recall. Additionally, our method demonstrated a minor difference of only 0.9% between precision and recall. In contrast, ROSE^2 [12] exhibited a 4.5% difference, and the Morphological-based method [3] had a 7.1% difference. This observation highlights that our method strikes a superior balance between over-segmentation and under-segmentation, in accordance with findings in [3]. This result can also be observed in the visualization result demonstrated in Fig. 3. Among ten selected scenes, "Colebrook" is the only scene devoid of furniture. Thus, by comparing the results obtained from "Colebrook" with those from other scenes, we can clearly observe the performance degradation experienced by previous methods in cluttered environments. It is shown in Table. I, precision and recall of previous methods drop from nearly 90% to less than 80% in cluttered settings, while our method exhibits stable performance across both non-cluttered and cluttered environments.

C. Exploration Performance

1) Test Scenes: To quantify the performance of exploration, following ANS [29], we evaluate all methods on 14 scenes from the Gibson validation set.

2) Baselines: To validate the exploration efficiency of our active room segmentation framework, we compare it against existing indoor exploration methods, including the original WFD [25], FFD [25], Gomez et al.'s method [23], and the learning-based ANS [29].

3) Results: We run experiments on a computer with Intel® Core™ i9-12900K CPU for each method with 50 different starting points in each scene. The averaged results are presented in Table. II. It is shown that Gomez et al.'s method [23] demonstrates the worst performance across all test scenes and WFD yields the best and the second-best performance in coverage area and ratio, respectively. Learning-based ANS shows superior efficiency compared to non-learning methods thanks to its short and consistent running time. Our method strikes a balance between the exploration completeness and efficiency, costing 100 seconds less compared to the original WFD whilst achieving a competitive coverage area of 31.33 m^2. It underscores the efficiency of the within-room exploration module and the constructed topology map in guiding the exploration process. The within-room exploration module enhances the efficiency of the frontier searching process by confining the searching area within one single room rather than the entire map. For the topology map, by breaking down the robot's trajectories into shorter segments with multiple waypoints, the framework reduces the likelihood of the robot becoming trapped in complex scenes, a common issue when navigating to distant goals in cluttered environments, and thus improves exploration efficiency and overall system robustness. Another interesting fact is that ANS achieves the highest coverage ratio among all comparing methods, but its corresponding coverage area is not the highest, suggesting that the exploration perfor-

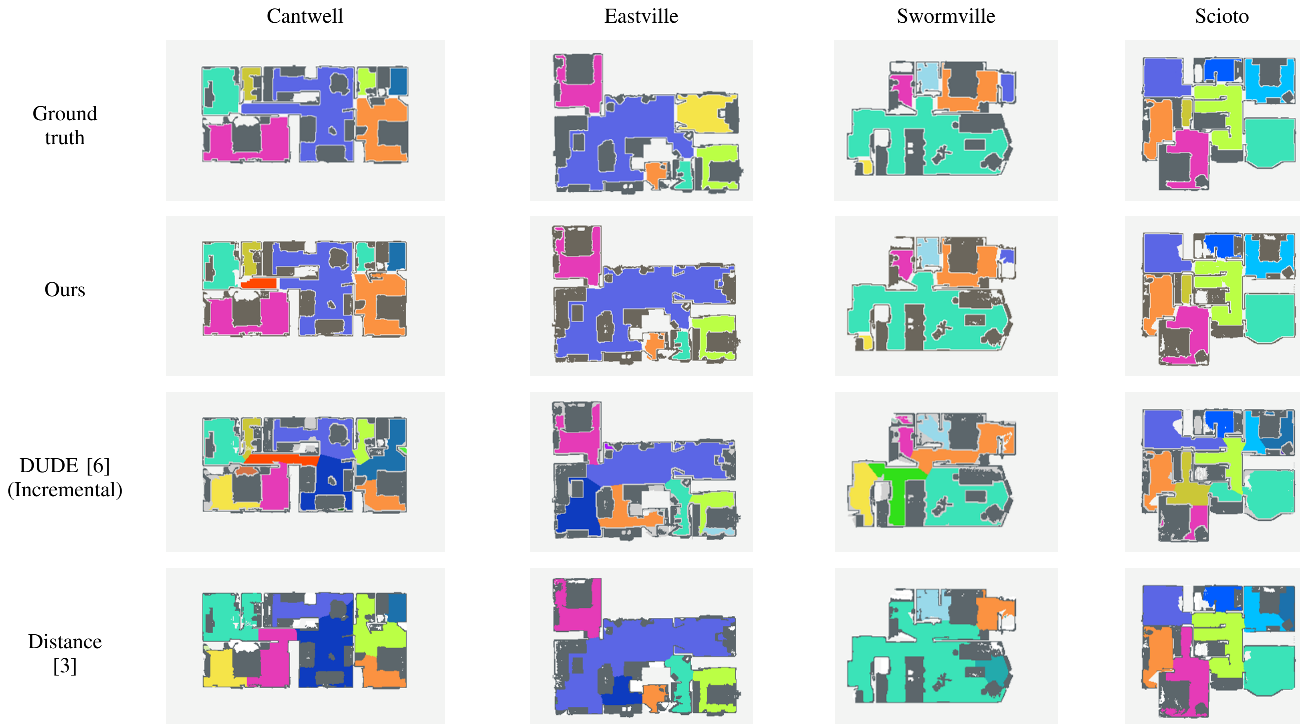


Fig. 3: Qualitative comparisons between our method and other approaches for active room semantic segmentation.

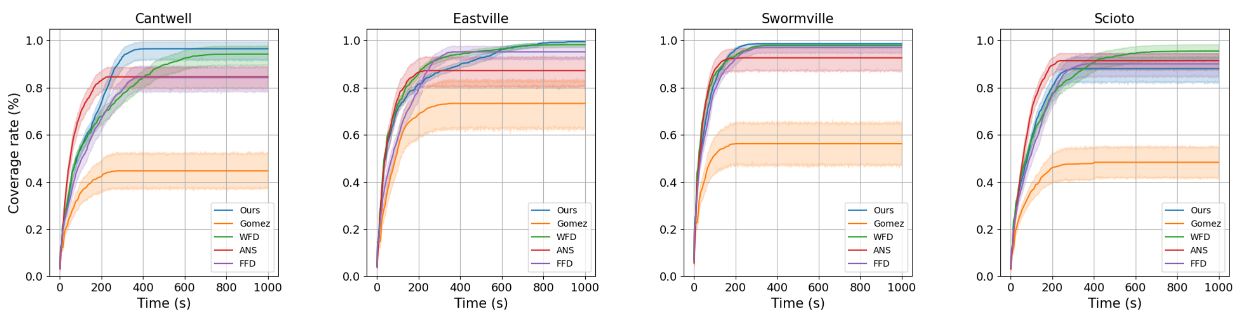


Fig. 4: The mean coverage-time curve and confidence interval of each exploration method on 4 test scenes.

TABLE II

EXPLORATION PERFORMANCE COMPARISON.

	Coverage ratio.(%)	Coverage area.(m^2)	Time(s)
Gomez	69.2	22.10	312
WFD	94.6	**31.83**	463
FFD	93.5	31.29	315
ANS	**94.9**	31.30	**229**
Ours	93.7	31.33	363

mance of ANS degrades as the size of the scenes grows. On the other hand, our method particularly yields superior performance for exploring large scenes with multiple rooms, as shown in Fig. 4, where results of "Cantwell," "Eastville," "Swormville," and "Scioto" are selected for visualization. On these four scenes, we achieved average coverage rates of 96.5%, 99.5%, 98.6%, and 88.1% respectively, compared to 84.6%, 87.2%, 92.6%, and 91.4% for ANS [29], 94.2%, 98.2%, 97.8%, and 95.5% for WFD [25], and 84.2%, 95.2%, 97.1%, and 90.0% for FFD [25]. These results conclusively illustrate that our exploration strategy not only preserves the original WFD's performance across almost all test scenes but also exhibits markedly enhanced efficiency, demonstrating the possibility of exploiting room segmentation results to facilitate the exploration process.

V. CONCLUSION

In this study, we have proposed an active room segmentation framework that enhances the robustness and autonomy of room segmentation by leveraging visual input and topological representation. Experimental results on the Gibson dataset have shown substantial improvement over previous methods in room segmentation and also demonstrate comparable performance in active exploration within cluttered indoor environments. However, this framework still has several limitations. The proposed topological representation is static and its quality highly depends on the door detection algorithm. Besides, choosing doors as the cue for room segmentation also makes the framework not generalizable to scenes where no obvious geometric boundaries exist between different functional areas. We acknowledge these limitations and are planning to solve all these issues in future research. Despite all these limitations, this framework represents a significant step forward in constructing and leveraging topological representations for room segmentation.

References

[1] R. Bormann, J. Hampp, and M. Hägele, "New brooms sweep clean-an autonomous robotic cleaning assistant for professional office cleaning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4470–4477.

[2] H. Myung, H.-m. Jeon, W.-Y. Jeong, and S.-W. Bang, "Virtual door-based coverage path planning for mobile robot," in *Advances in Robotics: FIRA RoboWorld Congress 2009, Incheon, Korea, August 16-20, 2009. Proceedings 12*. Springer, 2009, pp. 197–207.

[3] R. Bormann, F. Jordan, W. Li, J. Hampp, and M. Hägele, "Room segmentation: Survey, implementation, and analysis," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1019–1026.

[4] E. Fabrizi and A. Saffiotti, "Extracting topology-based maps from gridmaps," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3. IEEE, 2000, pp. 2972–2978.

[5] K. Joo, T.-K. Lee, S. Baek, and S.-Y. Oh, "Generating topological map from occupancy grid-map using virtual door detection," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–6.

[6] L. Fermin-Leon, J. Neira, and J. A. Castellanos, "Incremental contour-based topological segmentation for robot exploration," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2554–2561.

[7] J. Choi, M. Choi, and W. K. Chung, "Incremental topological modeling using sonar gridmap in home environment," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3582–3587.

[8] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson Env: real-world perception for embodied agents," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[9] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[10] A. Diosi, G. Taylor, and L. Kleeman, "Interactive slam using laser and advanced sonar," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1103–1108.

[11] M. Mielle, M. Magnusson, and A. J. Lilienthal, "A method to segment maps from different modalities using free space layout maoris: map of ripples segmentation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4993–4999.

[12] M. Luperto, T. P. Kucner, A. Tassi, M. Magnusson, and F. Amigoni, "Robust structure identification and room segmentation of cluttered indoor environments from occupancy grid maps," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7974–7981, 2022.

[13] Z. Zivkovic, B. Bakker, and B. Krose, "Hierarchical map building and planning based on graph partitioning," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 803–809.

[14] Y. Zhou, S. Yu, R. Sun, Y. Sun, and L. Sun, "Topological segmentation for indoor environments from grid maps using an improved njw algorithm," in *2017 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2017, pp. 142–147.

[15] M. Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3269–3274.

[16] O. M. Mozos and W. Burgard, "Supervised learning of topological maps using semantic information extracted from range data," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 2772–2777.

[17] F. Foroughi, J. Wang, A. Nemati, Z. Chen, and H. Pei, "Mapsegnet: A fully automated model based on the encoder-decoder architecture for indoor map segmentation," *IEEE Access*, vol. 9, pp. 101 530–101 542, 2021.

[18] P. Buschka and A. Saffiotti, "A virtual sensor for room detection," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 1. IEEE, 2002, pp. 637–642.

[19] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3491–3496.

[20] N. Sünderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile robot," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 5729–5736.

[21] L. Fermin-Leon, J. Neira, and J. A. Castellanos, "Tigre: Topological graph based robotic exploration," in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–6.

[22] M. Luperto, L. Fochetta, and F. Amigoni, "Exploration of indoor environments predicting the layout of partially observed rooms," *arXiv preprint arXiv:2004.06967*, 2020.

[23] C. Gomez, A. C. Hernandez, and R. Barber, "Topological frontier-based exploration and map-building using semantic information," *Sensors*, vol. 19, no. 20, p. 4595, 2019.

[24] 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'*. IEEE, 1997, pp. 146–151.

[25] M. Keidar and G. A. Kaminka, "Robot exploration with fast frontier detection: theory and experiments," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 2012, pp. 113–120.

[26] Z. Sun, B. Wu, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "Frontier detection and reachability analysis for efficient 2d graph-slam based active exploration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2051–2058.

[27] 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)*. IEEE, 2017, pp. 1396–1402.

[28] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[29] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," in *International Conference on Learning Representations*, 2020.

[30] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 400–418.

[31] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis, "Uncertainty-driven planner for exploration and navigation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 11 295–11 302.

[32] S. Dong, K. Xu, Q. Zhou, A. Tagliasacchi, S. Xin, M. Nießner, and B. Chen, "Multi-robot collaborative dense scene reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–16, 2019.

[33] S. S. Belavadi, R. Beri, and V. Malik, "Frontier exploration technique for 3d autonomous slam using k-means based divisive clustering," in *2017 Asia Modelling Symposium (AMS)*. IEEE, 2017, pp. 95–100.

[34] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.

[35] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.

[36] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

[38] K. Ye, S. Dong, Q. Fan, H. Wang, L. Yi, F. Xia, J. Wang, and B. Chen, "Multi-robot active mapping via neural bipartite graph matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 839–14 848.

[39] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv preprint arXiv:2004.05155*, 2020.

[40] M. Antonazzi, M. Luperto, N. Basilico, and N. A. Borghese, "Enhancing door detection for autonomous mobile robots with environment-specific data collection," *arXiv preprint arXiv:2203.03959*, 2022.