Real-time Localization and Mapping leveraging Hierarchical Representations

Hriday Bavle[1], Jose Luis Sanchez-Lopez[1], Muhammad Shaheer[1],
Javier Civera[2] and Holger Voos[1]

***Abstract*—In this paper, we present an evolved version of the Situational Graphs, which jointly models in a single optimizable factor graph a SLAM graph, as a set of robot keyframes, containing its associated measurements and robot poses, and a 3D scene graph, as a high-level representation of the environment that encodes its different geometric elements with semantic attributes and the relational information between those elements.**

Our proposed *S-Graphs+* is a novel four-layered factor graph that includes: (1) a keyframes layer with robot pose estimates, (2) a walls layer representing wall surfaces, (3) a rooms layer encompassing sets of wall planes, and (4) a floors layer gathering the rooms within a given floor level. The above graph is optimized in real-time to obtain a robust and accurate estimate of the robot's pose and its map, simultaneously constructing and leveraging the high-level information of the environment. To extract this high-level information, we present novel room and floor segmentation algorithms utilizing the mapped wall planes and free-space clusters.

We tested *S-Graphs+* on multiple datasets, including simulations of distinct indoor environments from real data captured over several construction sites and office environments, and on a real public dataset of indoor office environments. *S-Graphs+* outperforms relevant baselines in the majority of the datasets while extending the robot situational awareness by a four-layered scene model. Project web: https://snt-arg.github.io/s_graphs_docker/

I. Introduction

ROBOTS require a deep understanding of the situation for their autonomous and intelligent operations. Works like [1], [2], [3] generate 3D scene graphs modeling the environment with high-level semantic abstractions (such as chairs, tables, or walls) and their relationships (such as a set of walls forming a room or a corridor). While providing a rich understanding of the scene, they rely on separate SLAM methods, such as [4], [5], [6], that previously estimate the robot's pose and its map using metric/semantic representations without exploiting this hierarchical high-level information of the environment. Thus, in general, 3D scene graphs and their underlying SLAM graphs are not completely coupled.

*This work was partially funded by the Fonds National de la Recherche of Luxembourg (FNR), under the projects C19/IS/13713801/5G-Sky, by a partnership between the Interdisciplinary Center for Security Reliability and Trust (SnT) of the University of Luxembourg and Stugalux Construction S.A., by the Spanish Government under Grant PID2021-127685NB-I00 and by the Aragón Government under Grant DGA T45 17R/FSE. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

[1]Authors are with the Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. Holger Voos is also associated with the Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg. {hriday.bavle, joseluis.sanchezlopez, muhammad.shaheer, holger.voos}@uni.lu

[2]Author is with I3A, Universidad de Zaragoza, Spain jcivera@unizar.es

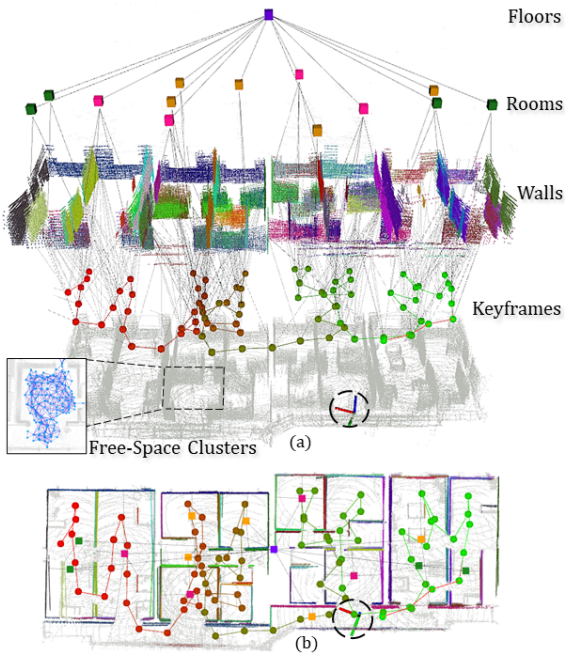


Fig. 1: *S-Graph+* built using a legged robot (circled in black) as it navigates a real construction site consisting of four adjacent houses. (a) 3D view of the four-layered hierarchical optimizable graph. The zoomed-in image shows a partial view of the free-space clusters utilized for room segmentation. (b) Top view of the graph.

Our previous work *S-Graphs* [7] bridges this gap proposing for the first time a tightly coupled geometric LiDAR SLAM with 3D scene graphs, demonstrating state-of-the-art metrics. However, it came with multiple limitations that we overcome in this work with our new *S-Graphs+* (Fig. 1), with updated front-end and back-end relying on 3D LiDAR measurements. Our main contributions are summarized as:

- A novel real-time factor graph organized in four hierarchical layers.
- A real-time extraction of high-level information using the novel room and floor segmentation algorithms.
- A thorough experimental evaluation in different simulated and real construction/office environments as well as software release for the research community.

II. Related Works

A. SLAM and Scene Graphs

The literature on LiDAR SLAM is huge, and there are several well-known geometric approaches like LOAM [4] and

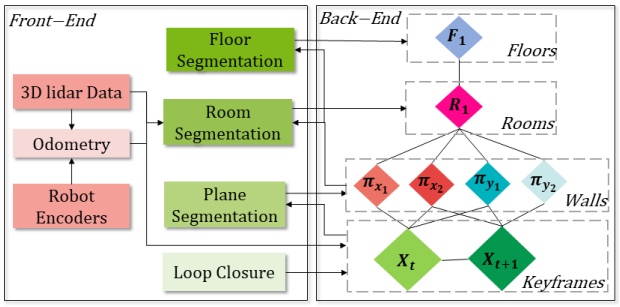


Fig. 2: *S-Graphs+* overview. Our inputs are the 3D LiDAR measurements and robot odometry, which are pre-filtered and processed in the front-end to extract wall planes, rooms, floor, and loop closures. Note the four-layered *S-Graph+*, whose parameters are jointly optimized in the back-end.

its variants [5], [8], [9], and also semantic ones like LeGO-LOAM [6], SegMap [10] that provide robust and accurate localization and 3D maps of the environments. While geometric SLAM lacks meaning in the representation of the environments, causing failures in aliased environments and limitations for high-level tasks or human-robot interaction, its semantic SLAM counterparts lack in most occasions geometric accuracy and robustness, due to wrong matches between the semantic elements and the limited relational constraints between them.

Scene graphs [2], [3] on the other hand, model scenes as structured representations, specifically in the form of a graph comprising objects, their attributes, and the inter-relationships among them. Though promising in terms of scene representation and higher-level understanding, a major drawback of these models is that they do not tightly couple the estimate of the scene graph with the SLAM state.

B. Room Segmentation

In the literature, different room segmentation techniques are presented over pre-generated maps using 3D LiDARs [11]–[13]. Their performance is, however, degraded in the presence of clutter. Authors in [3] present a real-time room segmentation approach to classifying different places into rooms but compared to our approach they do not utilize the walls in the environment to efficiently represent the rooms.

III. Overview

The architecture of *S-Graphs+* is illustrated in Fig. 2. Its pipeline can be divided into six modules, and its estimates are referred to four frames: the LiDAR frame L_t, the robot frame R_t, the odometry frame O, and the map frame M. L_t and R_t are rigidly attached to the robot and then depend on the time instant t, while O and M are fixed.

IV. Front-End

A. Wall Extraction

We use sequential RANSAC to detect and initialize wall planes. We refer the reader to [14] for further details.

B. Room Segmentation

It consists of two steps and the output is the parameters of **four-wall** and **two-wall rooms**.

Free-Space Clustering. Our free-space clustering algorithm divides the free-space graph of a scene into several clusters that should correspond to the rooms of that scene. Given a set of robot poses and a Euclidean Signed Distance Field (ESDF) representation [15] for these poses, we generate a sparsely connected graph $\mathcal{G}$ of free spaces using [16].

Given the graph $\mathcal{G}$, we cluster it into different free-space regions as follows. We create a filtered graph $\mathcal{G}_f$ removing the vertices $\boldsymbol{v}_d$ whose distance to obstacles is less than a given threshold t_λ. We also remove from $\mathcal{G}_f$ all the edges $\boldsymbol{e}_d$ that are connected to the node set $\boldsymbol{v}_d$. We then run the connected components method on $\mathcal{G}_f$ to divide it into several connected sub-graphs $\mathcal{G}_{f_i}, i \in \{1, \ldots, N\}$.

Room Extraction. Room extraction uses the free-space clusters $\mathcal{G}_{f_i}$ and the wall planes from a keyframe at time t to detect different room configurations. Wall planes are represented in the map frame, where each plane is defined by its normal and its distance to the origin. All extracted wall planes are first categorized as x-direction planes for which their highest normal component is n_x, and similarly y-direction planes. x-planes and y-planes are further classified into planes with positive and negative normal directions in x and y respectively. Given each sub-category of the wall planes, our room extraction method first checks the $L2$ norm between the 3D points of each plane and the vertices of each cluster $\mathcal{G}_{f_i}$, to find the set of walls lying closer to each specific cluster. If four plane candidates are found around the cluster we create a four-wall room center, and a two-wall room center is created using two-plane candidates. [14] presents room segmentation in detail.

C. Floor Segmentation

The floor segmentation module extracts the widest wall planes within the current explored floor level by the robot which can then be used to calculate the center of the current floor level. Our floor segmentation utilizes the information from all mapped walls to create a sub-category of wall planes as described in the room segmentation (Sec. IV-B). After receiving a complete plane set it computes the widths between all x-direction and y-direction planes. The wall plane set with the largest w_x and w_y are the chosen candidates for computing the center of the current floor level. [14] presents floor segmentation in detail.

V. Back-End

The back-end is responsible for creating and optimizing the four-layered *S-Graph+* summing the individual cost functions of each layer, explained in detail as follows.

Keyframes. This layer creates a factor node with the robot keyframe pose at time t in the map frame M. The pose nodes are constrained by pairwise odometry readings between consecutive poses as in [7].

Walls. This layer creates the planar factor nodes for the wall planes extracted by the wall segmentation (Sec. IV-A)

and constrain them with their corresponding keyframes using pose-plane constraints as in [7].

Rooms. The rooms layer receives the extracted room candidates and their corresponding wall planes from the room segmentation module (Sec. IV-B) to create appropriate constraints between them.

Four-Wall Rooms: We propose a novel edge formulation between the detected room node (generated from its center) and its four mapped wall planes, where the total cost function to minimize the room node and its plane set can be given as:

$$c_{\boldsymbol{\rho}}({}^{M}\boldsymbol{\rho}, \left[{}^{M}\boldsymbol{\pi}_{x_{a_i}}, {}^{M}\boldsymbol{\pi}_{x_{b_i}}, {}^{M}\boldsymbol{\pi}_{y_{a_i}}, {}^{M}\boldsymbol{\pi}_{y_{b_i}}\right]) = \sum_{t=1,i=1}^{T,S} \|{}^{M}\hat{\boldsymbol{\rho}}_i - f({}^{M}\tilde{\boldsymbol{\pi}}_{x_{a_i}}, {}^{M}\tilde{\boldsymbol{\pi}}_{x_{b_i}}, {}^{M}\tilde{\boldsymbol{\pi}}_{y_{a_i}}, {}^{M}\tilde{\boldsymbol{\pi}}_{y_{b_i}})\|^2_{\boldsymbol{\Lambda}_{\tilde{\boldsymbol{p}}_{i,t}}} \tag{1}$$

Where ${}^{M}\hat{\boldsymbol{\rho}}_i$ is the estimated four-wall room center obtained from Sec. IV-B and $f({}^{M}\tilde{\boldsymbol{\pi}}_{x_{a_i}}, {}^{M}\tilde{\boldsymbol{\pi}}_{x_{b_i}}, {}^{M}\tilde{\boldsymbol{\pi}}_{y_{a_i}}, {}^{M}\tilde{\boldsymbol{\pi}}_{y_{b_i}})$ is the function mapping the four wall planes estimated to a four-wall room center.

Two-Wall Rooms: We propose a similar cost function to minimize room nodes and their two corresponding wall planes as follows:

$$c_{\boldsymbol{\kappa}}({}^{M}\boldsymbol{\kappa}_i, \left[{}^{M}\boldsymbol{\pi}_{x_{a_1}}, {}^{M}\boldsymbol{\pi}_{x_{b_1}}, {}^{M}\mathbf{c}_i\right]) = \sum_{t=1,i=1}^{T,K} \|{}^{M}\hat{\boldsymbol{\kappa}}_i - f({}^{M}\tilde{\boldsymbol{\pi}}_{x_{a_1}}, {}^{M}\tilde{\boldsymbol{\pi}}_{x_{b_1}}, {}^{M}\mathbf{c}_i)\|^2_{\boldsymbol{\Lambda}_{\tilde{\boldsymbol{\kappa}}_{i,t}}} \tag{2}$$

${}^{M}\mathbf{c}_i$ is the cluster center, which is kept constant during the optimization, and ${}^{M}\hat{\boldsymbol{\kappa}}_i$ is the estimated two-wall room center in x direction obtained from Sec. IV-B. Duplicate wall plane nodes identified during the four-wall or two-wall room segmentation are constrained by a factor minimizing the difference between their respective parameters.

Floors. The floor node consists of the center of the current floor level calculated from the floor segmentation (Sec. IV-C). We add the relative position cost function between the floor node and all the mapped four-wall and two-wall rooms node at that floor level.

VI. Experimental Results

A. Methodology

S-Graphs+ is built on top of its baseline *S-Graphs* [7] and is validated on simulated and real-world scenarios, comparing it against several state-of-the-art LiDAR SLAM frameworks and its baseline. The experiments cover a wide array of scenes, from construction sites to office spaces, and use data recorded in-house and from the public TIERS dataset.

Simulated Data. We conduct a total of five simulated experiments in indoor environments with different room configurations. Due to absence of odometry from robot encoders, in all simulated experiments the odometry is estimated only from LiDAR measurements. For a fair validation, *S-Graphs+* is run using two different odometry inputs, specifically VGICP [17] and FLOAM [5]. Tab. I showcases the ATE for the simulated experiments. We outperform *S-Graphs* [7] thanks to the new plane segmentation module, new rooms factors and the new room segmentation algorithm. Experiments are sorted by scene size, from left to right the scene size being larger. Note how the baseline errors tend to grow for larger scenes, and how our *S-Graphs+* achieves bigger error reductions for larger scenes due to its better representation.

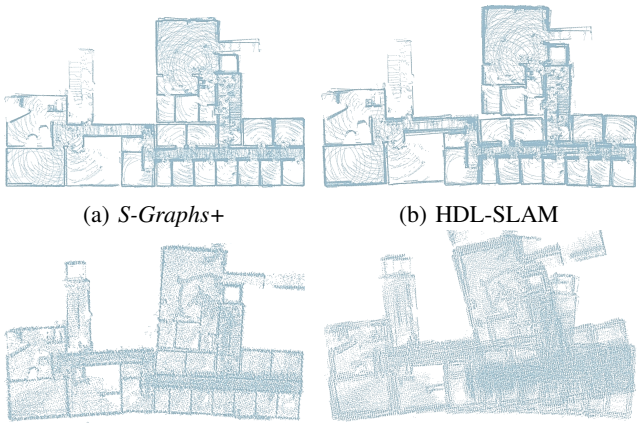
TABLE I: Absolute Trajectory Error (ATE) [m], of *S-Graph+* and relevant baselines on simulated data. Best results are bold-faced, second best are underlined. '-' refers to an unsuccessful run.

Method		**Dataset** (m x 10^{-2})				
Mapping	Odometry	*C1F0*	*C1F2*	*SE1*	*SE2*	*SE3*
HDL-SLAM [9]	VGICP [17]	9.42	<u>2.12</u>	2.46	10.6	6.23
ALOAM [4]	ALOAM	9.90	8.70	15.7	40.2	19.7
MLOAM [8]	MLOAM	-	50.2	66.1	-	15.7
FLOAM [5]	FLOAM	11.7	14.5	14.6	30.5	27.6
LeGO-LOAM [6]	LeGO-LOAM	-	-	-	-	74.1
S-Graphs [7]	VGICP	5.09	2.57	<u>2.18</u>	**9.10**	<u>3.86</u>
S-Graphs+	VGICP	**4.47**	**1.75**	**1.91**	<u>9.31</u>	**3.37**
S-Graphs+	FLOAM	5.94	11.7	5.72	9.60	19.9

In-House Dataset. In all our in-house data we utilize the robot encoders for estimating the odometry. The experiments from *C1F1-C4F1* are performed over construction sites with different room layouts. *LC1* consists of an office environment in which the robot traverses back and forth a long corridor.

Tab. IV presents the point cloud RMSE obtained by comparing the generated 3D maps against the 3D maps from the building plans. As it can be observed in the table, *S-Graphs+* is more accurate than the baseline in most of the cases. For experiment *C4F0*, Fig. 3 shows a top view of the final maps estimated by *S-Graphs+* and three other baselines. Observe the higher degree of accuracy and cleaner map elements in the *S-Graphs+* case. Additionally, Tab. III provides a comprehensive overview of the computation time required by each module within *S-Graphs+*, as can be seen, even for experiments with approximately 17 mins (*C2F2*), all the modules of *S-Graphs+* are able to maintain real-time performance.

TIERS LiDARs dataset. We also validate *S-Graphs+* on



(a) *S-Graphs+* (b) HDL-SLAM

(c) ALOAM (d) FLOAM

Fig. 3: Maps by *S-Graphs+* and baselines, in-house seq. *C4F0*.

TABLE IV: Point cloud RMSE [m] for our in-house real sequences. All methods use odometry from robot encoders. Best results are boldfaced, second best underlined. '-' refers to an unsuccessful run.

	Dataset (m x 10^{-2})						
Method	**Point Cloud RMSE**						
Mapping	*C1F1*	*C1F2*	*C2F0*	*C2F1*	*C2F2*	*C3F1*	*C3F2*
HDL-SLAM [9]	33.5	<u>19.8</u>	18.5	<u>21.1</u>	<u>19.5</u>	22.9	<u>19.4</u>
ALOAM [4]	52.6	33.6	34.1	45.1	29.9	36.5	43.4
MLOAM [8]	45.0	27.6	40.6	32.4	23.6	-	-
FLOAM [5]	68.5	39.2	40.2	55.5	39.5	58.3	38.8
LeGO-LOAM [6]	-	-	39.2	45.5	-	52.9	50.3
S-Graphs [7]	**33.1**	**18.9**	18.4	21.8	**17.6**	<u>22.8</u>	22.6
S-Graphs+	<u>32.9</u>	**18.9**	**16.9**	**18.9**	**17.6**	**22.3**	**18.7**

TABLE II: Absolute Trajectory Error (ATE) [m], of *S-Graphs+* and relevant baselines on the TIERS dataset [18]. Best results boldfaced, second best underlined.

Method		**Dataset** (m x 10^{-2})				
Mapping	Odometry	*T6*	*T7*	*T8*	*T10*	*T11*
HDL-SLAM [9]	VGICP [17]	<u>25.6</u>	27.3	**31.0**	148.9	287.1
ALOAM [4]	ALOAM	25.7	27.0	34.6	<u>68.1</u>	234.9
MLOAM [8]	MLOAM	25.7	**26.1**	33.9	263.4	**47.4**
FLOAM [5]	FLOAM	25.8	<u>26.3</u>	32.4	71.3	161.1
LeGO-LOAM [6]	LeGO-LOAM	27.3	33.5	36.3	140.9	68.2
S-Graphs [7]	VGICP	25.6	26.8	35.1	260.1	190.1
S-Graphs+	VGICP	25.6	26.6	32.9	126.6	162.3
S-Graphs+	FLOAM	**25.2**	26.5	<u>32.1</u>	**48.3**	<u>60.6</u>

TABLE III: Computation time [ms] of *S-Graphs+* along the total length of the sequence [s] for In-House dataset.

	Dataset							
	Computation Time (mean) [ms]							
Module	*C1F1*	*C1F2*	*C2F0*	*C2F1*	*C2F2*	*C3F1*	*C3F2*	*LC1*
Plane Segmentation	91.8	47.4	68.9	45.3	82.2	57.6	44.8	80.0
Room Segmentation	17.6	9.8	9.6	5.3	10.7	2.9	4.2	7.1
Floor Segmentation	8.1	3.4	4.6	7.2	44.7	4.0	16.9	9.6
Back-End	74.0	105.7	87.3	169.0	263.1	124.5	173.2	85.2
Sequence Length [s]	487	657	238	672	1044	558	999	339

the public TIERS dataset [18]. Experiments *T6* to *T8* are done in a single small room in which the platform does several passes at increasing speeds. Experiments *T10* and *T11* are performed in a larger indoor hallway with longer trajectories of the moving platform. Due to the absence of encoder readings in this dataset, each baseline method uses its own LiDAR-based odometry.

Tab. II presents the ATE for all baseline methods and our *S-Graphs+*. *S-Graphs+* with FLOAM odometry gives the best results in all the experiments. Again, the sequences are sorted from left to right by increasing size. Note that all methods perform similarly for small scenes, but differ as scenes become larger, *S-Graphs+* presenting significant error reductions for large environments. The strength of our hierarchical representation is particularly evident in scenarios like *T11*, in which *S-Graphs+* keeps the errors small even if the FLOAM odometry error grows substantially.

VII. Conclusion

In this work we present *S-Graphs+*, a novel four-layered hierarchical factor graph composed of a *keyframes layer*, *walls layer*, *rooms layer* and *floors layer*. To extract this high-level information we also propose a novel room segmentation algorithm using free-space clusters and wall planes and a floor segmentation algorithm extracting the floor centers using all the currently extracted wall planes. We demonstrate state-of-the-art results against several baselines on simulated and real experiments covering different office and construction indoor environments.

References

[1] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, "3D Scene Graph: A structure for unified semantics, 3D space, and camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5664–5673.

[2] S.-C. Wu, J. Wald, K. Tateno, N. Navab, and F. Tombari, "Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7515–7525.

[3] N. Hughes, Y. Chang, and L. Carlone, "Hydra: A Real-time Spatial Perception Engine for 3D Scene Graph Construction and Optimization," *arXiv preprint arXiv:2201.13360*, 2022.

[4] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014.

[5] H. Wang, C. Wang, C. Chen, and L. Xie, "F-LOAM : Fast LiDAR Odometry and Mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[6] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765.

[7] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "Situational graphs for robot navigation in structured indoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9107–9114, 2022.

[8] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 351–371, 2021.

[9] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, Mar. 2019.

[10] R. Dubé , A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, jul 2019.

[11] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D Semantic Parsing of Large-Scale Indoor Spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1534–1543.

[12] R. Ambruş, S. Claici, and A. Wendt, "Automatic Room Segmentation From Unstructured 3-D Data of Indoor Environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 749–756, 2017.

[13] S. Ochmann, R. Vock, and R. Klein, "Automatic reconstruction of fully volumetric 3D building models from oriented point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 151, pp. 251–262, 2019.

[14] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "S-graphs+: Real-time localization and mapping leveraging hierarchical representations," 2023.

[15] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. 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)*. IEEE, sep 2017.

[16] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Sparse 3d topological graphs for micro-aerial vehicle planning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[17] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for Fast and Accurate 3D Point Cloud Registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 054–11 059.

[18] L. Qingqing, Y. Xianjia, J. P. Queralta, and T. Westerlund, "Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3837–3844.