Towards Localizing Structural Elements: Merging Geometrical Detection with Semantic Verification in RGB-D Data

Ali Tourani[1], Saad Ejaz[1], Hriday Bavle[1], Jose Luis Sanchez-Lopez[1], and Holger Voos[1]

***Abstract*— RGB-D cameras supply rich and dense visual and spatial information for various robotics tasks such as scene understanding, map reconstruction, and localization. Integrating depth and visual information can aid robots in localization and element mapping, advancing applications like 3D scene graph generation and Visual Simultaneous Localization and Mapping (VSLAM). While point cloud data containing such information is primarily used for enhanced scene understanding, exploiting their potential to capture and represent rich semantic information has yet to be adequately targeted. This paper presents a real-time pipeline for localizing building components, including wall and ground surfaces, by integrating geometric calculations for pure 3D plane detection followed by validating their semantic category using point cloud data from RGB-D cameras. It has a parallel multi-thread architecture to precisely estimate poses and equations of all the planes detected in the environment, filters the ones forming the map structure using a panoptic segmentation validation, and keeps only the validated building components. Incorporating the proposed method into a VSLAM framework confirmed that constraining the map with the detected environment-driven semantic elements can improve scene understanding and map reconstruction accuracy. It can also ensure (re-)association of these detected components into a unified 3D scene graph, bridging the gap between geometric accuracy and semantic understanding. Additionally, the pipeline allows for the detection of potential higher-level structural entities, such as rooms, by identifying the relationships between building components based on their layout.**

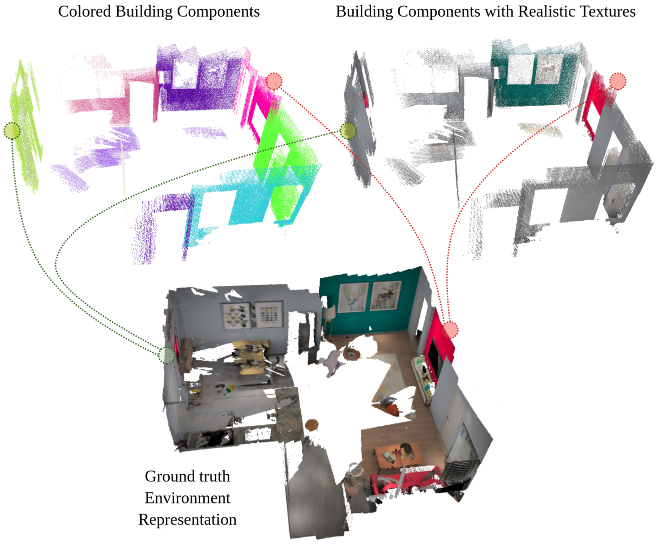

Colored Building Components
Building Components with Realistic Textures
Ground truth
Environment
Representation

Fig. 1: The 3D reconstructed map of an indoor environment using the proposed approach with **(left)** distinct color-coded building components for clear distinction and **(right)** the original RGB textures for a realistic representation.

I. Introduction

Employing the rich, dense data captured by sensors like Light Detection And Ranging (LiDAR) or RGB-D cameras enables robots to effectively address challenges in environment understanding, reconstruction, and Simultaneous Localization and Mapping (SLAM) [1]. The point clouds generated by these sensors provide detailed spatial information, facilitating precise detection, localization, and 3D mapping of the environment elements essential for robotic perception [2]. In this context, RGB-D vision sensors offer a more comprehensive understanding of environments by fusing color and depth information, compensating for monocular cameras' lack of depth perception and LiDAR sensors' limited visual richness.

As vision sensors provide a versatile and cost-efficient strategy for map reconstruction, they established a unique extension of SLAM algorithms titled Visual SLAM (VSLAM) [3]. Computer vision methodologies are inevitable in VSLAM, as they provide rich scene understanding and semantic object localization [4], [5]. Such visual data can also form the scanned environment into optimizable 3D scene graphs and depict their representations in multi-level abstraction models [6], [7]. In both cases, providing integrated depth and visual information further enhances the robot's localization and mapping of the observed elements.

Given the advantages of point clouds generated by RGB-D cameras, this paper aims to employ combined depth and visual information for geometrical calculation and semantic confirmation, respectively. The pipeline initiates with plane extraction using the depth data provided in the point clouds of a given frame and completes with semantic label augmentation through processing its visual data, resulting in

[1]Authors are with the Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg, Luxembourg. Ali Tourani is also associated with the Institute for Advanced Studies (IAS), University of Luxembourg, Luxembourg. Holger Voos is also associated with the Faculty of Science, Technology, and Medicine, University of Luxembourg, Luxembourg. {ali.tourani, saad.ejaz, hriday.bavle, joseluis.sanchezlopez, holger.voos}@uni.lu

*This research was funded, in whole or part, by the Luxembourg National Research Fund (FNR), DEUS Project, ref. C22/IS/17387634/DEUS. In addition, it was partially funded by the Institute of Advanced Studies (IAS) of the University of Luxembourg through an "Audacity" grant (project TRANSCEND - 2021).

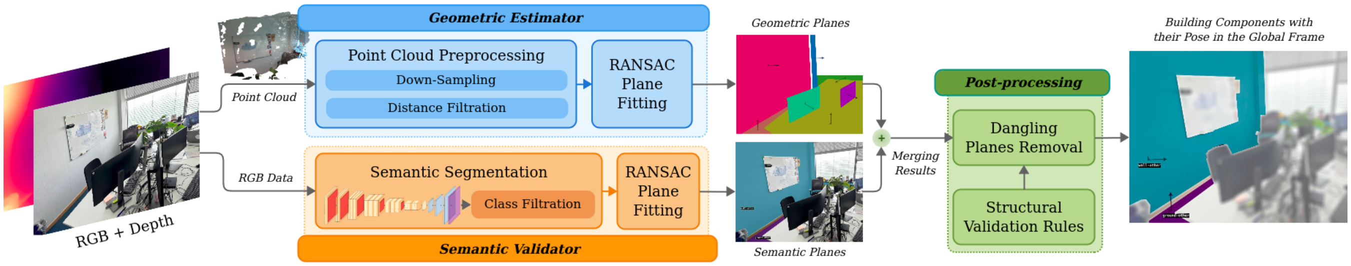


Fig. 2: The order of steps to detect building components from RGB-D data using the proposed approach.

rich information for building component localization. Fig. 1 depicts the outcome of the proposed methodology in accurate mapping and scene understanding. Accordingly, the primary contributions of the paper are summarized below:

- A real-time pipeline for accurate recognition and localization of building-level semantic components, *i.e.,* wall and ground surfaces,
- A method for (re-)association of the detected building components and representing them in 3D scene graphs; and
- Incorporating the mentioned rich semantic entities to enhance the performance of VSLAM through imposing environment-driven constraints.

II. Related Works

The implication of localizing geometric and semantic entities is highly applicable in robotics and covers various fields of study. In this regard, visual localization, feature matching, and pose estimation lay the foundation for navigation and mapping tasks, *e.g.,* VSLAM. Taira *et al.* [8] introduced a methodology for proper pose verification of visited objects from different camera views by combining appearance, geometry, and semantics modalities. Peng *et al.* [9] presented a voxel-to-point algorithm to construct semantic and geometric features of given point clouds for outdoor environments. In [10], semantic categories are assigned to point cloud data by extensively exploring geometric features extracted using a multi-geometry and boundary-guided encoder-decoder. Despite their effectiveness, the introduced methods and many others [11]–[13] have not integrated with SLAM procedures for extensive map building and real-time localization.

On the other hand, enhancing VSLAM applications with semantic and geometric constraints has shown potential for improving accuracy and scene understanding [14]. Accordingly, incorporating geometric components such as planes [15], [16] and lines [17], [18] helps reduce the environment's dimensionality, leading to more precise map reconstruction. Alternatively, localizing and mapping semantic entities by estimating their poses using point cloud processing [19]–[24] or obtaining the pose information from fiducial markers attached to them [25], [26] offers accurate positioning and identification of critical environmental elements. It can be seen that combining geometric and semantic constraints aids VSLAM systems in achieving better scene comprehension and mapping of the detected entities.

The approach proposed in this paper pinpoints the gap between geometric object localization and semantic class verification within a unified framework, leading to accurate building component localization. Filling this gap makes it possible to detect potential structural entities (*e.g.,* rooms) using building components (*i.e.,* wall and ground surfaces) regarding their layout and inner relationships, leading to the reconstruction of higher-accuracy maps.

III. Proposed Method

Fig. 2 depicts the pipeline of the proposed methodology. The process begins with a sequence of video frames captured by RGB-D cameras, providing colored visual data and point clouds enriched with depth information. The pipeline can be optimized for VSLAM by processing keyframes instead of every individual frame, reducing computational costs while maintaining the real-time performance for localization and mapping. The point cloud and RGB data are passed to the "*geometric estimator*" and "*semantic validator*" modules, respectively, both running in parallel on separate threads. The different stages of the pipeline are further detailed in the following subsections, where each module's functionality and part are explained:

A. Geometric Estimator

This module processes the point cloud data captured by the RGB-D sensor. Its primary objective is to investigate the points within the cloud to detect and extract potential 3D planes (*i.e.,* planar regions) present in the environment. Despite the point cloud's dense and rich spatial information, it may contain noisy or low-resolution points, negatively impacting subsequent processing steps. Thus, the given point cloud $\mathbf{P} = \{\mathbf{p}_i \mid i = 1, 2, \ldots, n\}$ consisting of points $\mathbf{p} = (x, y, z, r, g, b)$ undergoes a two-stage preprocessing procedure. In the first step, down-sampling is applied to select $\mathbf{P_d} = \{\mathbf{p}_i \mid i = 1, 2, \ldots, m\}$, where $n \leq m$ and $\mathbf{P_d} \subset \mathbf{P}$. Next, distance filtering is employed to only keep the 3D points that satisfy the condition $\mathbf{P_f} = \{\mathbf{p}_i \in \mathbf{P_d} \mid \theta_{min} \leq \|\mathbf{p}_i\| \leq \theta_{max}\}$, where θ_{min} and θ_{max} are minimum and maximum desired depth ranges, respectively. This results in the final preprocessed point cloud $\mathbf{P_f}$.

The next step is to apply RANdom SAmple Consensus (RANSAC) [27] plane fitting algorithm on $\mathbf{P_f}$ to detect potential planes with their geometric equations. Hence, a set of random points $\mathbf{p}_{\mathbf{r}_i} \in \mathbf{P_f}$ are iteratively selected to calculate the normal vectors $\mathbf{n}_j \in \mathbb{R}^3$ $(j \in \mathbb{N})$ representing a set of planes π_j with the pre-defined distance $d(\mathbf{p_r}_i, \pi_j) \leq \epsilon$, where ϵ is the inlier threshold. The final output consists of the best-fitted geometric planes $\mathbf{\Pi} = \{\pi_j \mid \pi_j \in \mathbb{R}^3, j \in \mathbb{N}\}$, representing all objects with planar surfaces detected from the filtered point cloud $\mathbf{P_f}$. It is essential to mention that the detected planes may correspond to various objects, such as tables, floors, screens, *etc.* (as shown in Fig. 2), many of which are irrelevant for detecting building components required for understanding the environment's layout.

B. Semantic Validator

In parallel to the *geometric estimator*, RGB frames are passed into the *semantic validator* module to extract semantic information, explicitly pivotal building components, including walls and ground surfaces. This module applies semantic segmentation to label each pixel in the RGB data, ensuring that only relevant classes are kept while irrelevant entities are discarded. Panoptic segmentation is preferred in this context due to its better occlusion handling, sharper boundary detection, and efficient distinguishing among objects within the same class. Accordingly, the proposed method employs real-time panoptic semantic segmentation frameworks, including PanopticFCN [28] and YOSO [29].

Each frame $f \in \mathbf{F}$ passes through the semantic segmentation function S, resulting in the semantic label map $L = S(f)$, where $L_{(x,y)}$ refers to the semantic class label of the pixel at $(x, y) \in \mathbb{N}^2$. The next step is to apply semantic *class filtration* to filter building components' semantic classes $\mathbf{\Psi}$ (*i.e.,* wall and ground surfaces) as follows:

$$L_{\Psi(x,y)} = \begin{cases} L_{(x,y)} & \text{if } L_{(x,y)} \in \Psi \\ \textit{null} & \textit{otherwise} \end{cases}$$

where L_Ψ represents the filtered semantic label map derived from the input RGB frame, containing only the pixels corresponding to the building components specified in $\mathbf{\Psi}$. The uncertainty parameter $\lambda_{(x,y)} \in \mathbb{R}$ for each pixel $L_{\Psi(x,y)}$ is also stored for potential classification errors.

Since the semantic map L_Ψ may contain noisy and incomplete segments, it requires further refinement. Therefore, another pass of RANSAC is applied to L_Ψ to filter out noisy and incomplete parts, considering the uncertainty parameter λ. The outcome is referred to as $\mathbf{\Pi}_\Psi = \{\pi_{\psi_k} \mid \pi_{\psi_k} \in \mathbb{R}^3, k \in \mathbb{N}\}$ and represents all building component objects derived from the L_Ψ. It should be noted that $\mathbf{\Pi}_\Psi$ may suffer from sparsity due to segmentation resolution and occlusion constraints, making its geometric equations inconsistent. An instance of the extracted labeled planar entities using the *semantic validator* module is shown in Fig. 2.

C. Geometric-Semantic Fusion

The geometric planes $\mathbf{\Pi}$ estimated by the *geometric estimator* are more trustworthy due to being derived from dense point cloud data, delivering accurate geometric analyses. However, these planes lack semantic information and can belong to any semantic entity with planar surfaces. Conversely, the planes $\mathbf{\Pi}_\Psi$ determined by the *semantic validator* contain semantic labels belonging to specific building components, but they lack precise geometric equations. Hence, the final stage of the pipeline involves associating $\mathbf{\Pi}$ with $\mathbf{\Pi}_\Psi$ planes belonging to frame f_t captured at time t, enabling the validation of the geometric planes and assigning accurate semantic labels to them. The essence of $\mathbf{\Pi}_\Psi$ planes in the fusion stage is to validate the semantic labeling of $\mathbf{\Pi}$ planes after matching, ensuring that the matched planes possess accurate geometric properties and are correctly identified as pre-defined building elements.

Considering geometric plane $\pi_i \in \mathbf{\Pi}$ and semantic plane $\pi_{\psi_j} \in \mathbf{\Pi}_\Psi$, the association process is defined as follows:

$$\pi_i = \begin{cases} \pi_{\psi_j} & \text{if } \textit{match}(\pi_i, \pi_{\psi_j}) \geq \epsilon \\ \textit{null} & \textit{otherwise} \end{cases}$$

where ϵ is a predefined threshold for a valid match with the most prominent inlier matches. The matched (validated) planes are stored in $\mathbf{\Pi}_\kappa \subset \mathbf{\Pi}$, which contain geometrically accurate and semantically meaningful planes.

The final step is applying post-processing algorithms on $\mathbf{\Pi}_\kappa$. Accordingly, the dangling planes removal function $\mathbf{\Pi}_\kappa = D(\mathbf{\Pi}_\kappa)$ omits any planes of $\mathbf{\Pi}_\kappa$ that do not correspond to wall or ground surfaces. The remaining valid planes are passed through a structural validation function $\mathbf{\Pi}_\kappa = V(\mathbf{\Pi}_\kappa)$, ensuring that they satisfy the necessary geometric constraints for the environment. For instance, ground planes are constrained to be horizontal in the global reference frame, while wall planes must be vertical and perpendicular to the ground surfaces. Detecting such building components prepares the ground for identifying high-level structural-semantic elements, such as rooms and corridors, defined by the topological associations among the mentioned elements.

IV. Evaluation

This section describes the two assessments conducted to evaluate the proposed methodology's accuracy and usefulness. One experiment analyzes the correct recognition of building components in the environment, while the second confirms its impact on map reconstruction by integrating with a VSLAM framework.

A. Evaluation Setup

The evaluations were conducted on a system equipped with an 11^{th} Gen. Intel® Core™ i9-11950H processor running at 2.60 GHz, an NVIDIA T600 Mobile GPU with 4GB of GDDR6 memory, and 32GB of RAM. The employed datasets were ICL [30], ScanNet [31], and an in-house dataset collected by the authors in real-world environments. Table I presents the characteristics of the mentioned datasets used for evaluation.

It should be noted that the in-house dataset contains sequences recorded in various indoor setups using a designed handheld/mountable device equipped with an *Intel® RealSense™ Depth Camera D435i* and an *Ouster OS1 Mid-Range High-Resolution Imaging LiDAR* with the precision of $0.3 - 1m : \pm 0.7cm$. The LiDAR data is primarily used

TABLE I: The list of datasets used for evaluations.

Dataset	*#Instances*	*Duration*	*Texturing*
ICL [30]	09	11m 26s	photo-realistic
ScanNet [31]	06	04m 32s	real
In-house (collected)	07	41m 36s	real
Total	**22**	**57m 34s**	

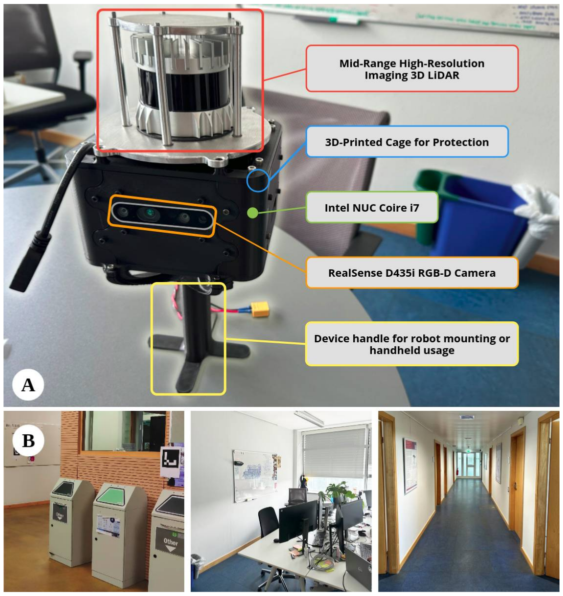

Mid-Range High-Resolution Imaging 3D LiDAR
3D-Printed Cage for Protection
Intel NUC Coire i7
RealSense D435i RGB-D Camera
Device handle for robot mounting or handheld usage
A
B

Fig. 3: The in-house data collection setup **(A)** and some instances of the collected dataset **(B)**.

as ground truth. Fig. 3 depicts the device employed for collecting data and some RGB frames from our in-house dataset.

B. Map Reconstruction Performance

To showcase the effectiveness of the proposed approach within a VSLAM framework, we integrated our pipeline equipped with YOSO [29] semantic segmentation into ORB-SLAM 3.0 [32]. This integration detects and maps the defined building components (in this work wall and ground surfaces), adding additional constraints while preserving the system's real-time performance. The evaluation results are presented in Table II, where the baseline (ORB-SLAM 3.0) and its pipeline-integrated version were executed across eight iterations on the datasets to ensure reliable outcomes. The employed evaluation metric is Root Mean Square Error (RMSE) in *meters*.

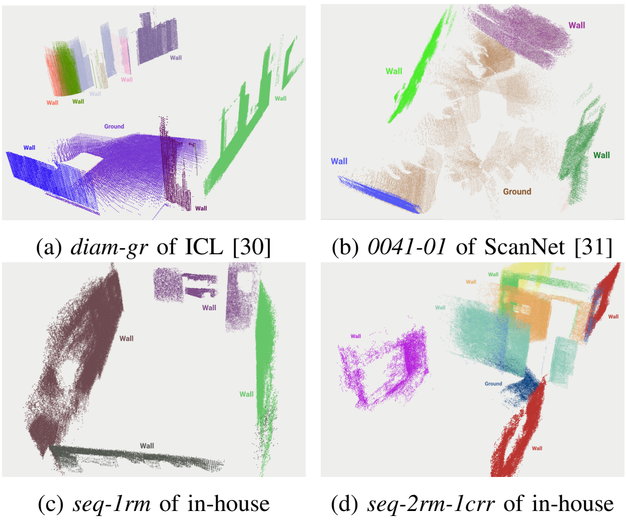
As shown in Table II, the pipeline-integrated version of ORB-SLAM 3.0 outperforms the baseline across various environment layouts in most of the cases. This improvement is primarily due to incorporating environment-specific constraints derived from accurately detecting building components, including wall and ground surfaces. With such constraints, the system achieves a more structured and informed reconstructed map, reducing trajectory estimation errors and better aligning with real-world features. In the ICL dataset, where the environment is relatively small, the most significant improvements -around 30%- occurred in *diam-gr* and *diam-wh*, with reductions of *2.9 cm* and *1.2 cm* in RMSE, respectively. Sequence *seq-0200-00* in the ScanNet dataset leads to a higher trajectory error of *0.6 cm*, presumably due to errors in associating building components after getting lost and incomplete map reconstruction affecting overall accuracy. In the collected real-world in-house dataset where the sequences are longer, the best results appear in *seq-1rm-1crr* (one office connected to a corridor) and *seq-4crr* (four connected corridors) with *5.8%* and *5.6%* improvements, respectively. Some instances of the reconstructed maps presented in 3D scene graphs are depicted in Fig. 4.

TABLE II: Evaluation results on the various dataset using Root Mean Square Error (RMSE) in *meters*. The best results are highlighted in bold. Numbers in brackets indicate the percentage of the traversed trajectory before tracking failure, with *cmp.* denoting completion.

		Methodologies		**Diff.(%)**
Dataset	**Sequence**	*ORB-SLAM 3.0*	*Proposed*	
ICL [30]	deer-gr	0.008 $^{[58\%]}$	**0.007** $^{[58\%]}$	+12.5%
	deer-wh	0.068 $^{[53\%]}$	0.068 $^{[53\%]}$	0.0%
	deer-w	0.098 $^{[55\%]}$	**0.094** $^{[61\%]}$	+4.1%
	deer-r	0.065 $^{[64\%]}$	**0.064** $^{[64\%]}$	+1.5%
	deer-mavf	0.027 $^{[cmp.]}$	**0.026** $^{[cmp.]}$	+3.7%
	diam-gr	0.097 $^{[cmp.]}$	**0.068** $^{[cmp.]}$	+29.9%
	diam-wh	0.041 $^{[92\%]}$	**0.029** $^{[cmp.]}$	+29.3%
	diam-mavs	**0.017** $^{[cmp.]}$	0.018 $^{[cmp.]}$	-5.9%
	diam-mavf	0.027 $^{[cmp.]}$	**0.026** $^{[cmp.]}$	+3.7%
ScanNet [31]	seq-0041-01	0.141 $^{[cmp.]}$	**0.136** $^{[95\%]}$	+3.5%
	seq-0103-01	0.077 $^{[39\%]}$	**0.076** $^{[39\%]}$	+1.3%
	seq-0200-00	**0.047** $^{[61\%]}$	0.052 $^{[61\%]}$	-10.6%
	seq-0560-00	0.090 $^{[94\%]}$	**0.092** $^{[cmp.]}$	-2.2%
	seq-0614-01	**0.126** $^{[72\%]}$	0.132 $^{[69\%]}$	-4.8%
	seq-0626-00	0.095 $^{[cmp.]}$	**0.094** $^{[cmp.]}$	+1.1%
Collected (in-house)	seq-1rm	0.063 $^{[cmp.]}$	0.063 $^{[cmp.]}$	0.0%
	seq-4rm-2crr	0.432 $^{[cmp.]}$	**0.418** $^{[cmp.]}$	+3.2%
	seq-nested	**0.163** $^{[88\%]}$	0.169 $^{[98\%]}$	-3.7%
	seq-1rm-1crr	0.278 $^{[cmp.]}$	**0.262** $^{[cmp.]}$	+5.8%
	seq-4crr	0.285 $^{[75\%]}$	**0.269** $^{[75\%]}$	+5.6%
	seq-2rm-1crr	**0.125** $^{[cmp.]}$	0.126 $^{[cmp.]}$	-0.8%
	seq-lab	**0.376** $^{[99\%]}$	0.377 $^{[cmp.]}$	-0.3%

C. Building Component Recognition Accuracy

Table III presents the performance of the proposed approach in accurately detecting building components across different datasets. For evaluation, we calculated the values of True Positive (TP), False Positive (FP), and False Negative (FN) for each dataset instance. In this context, a TP represents a building component that exists in the environment and is correctly detected by the method. A FN occurs when a building component is incorrectly detected where none exists, while an FP indicates a failure to detect an existing component. Accordingly, the proposed method's performance was assessed using object-level precision, recall, and the F1-Score calculations. While precision measures the ratio of correctly detected components relative to all detected ones, recall reflects the method's ability to detect all existing building components, making it possible to provide



(a) *diam-gr* of ICL [30] (b) *0041-01* of ScanNet [31]

(c) *seq-1rm* of in-house (d) *seq-2rm-1crr* of in-house

Fig. 4: Reconstructed maps presented in 3D scene graphs enriched with recognized building components in some dataset instances. Labels have been manually added to enhance clarity.

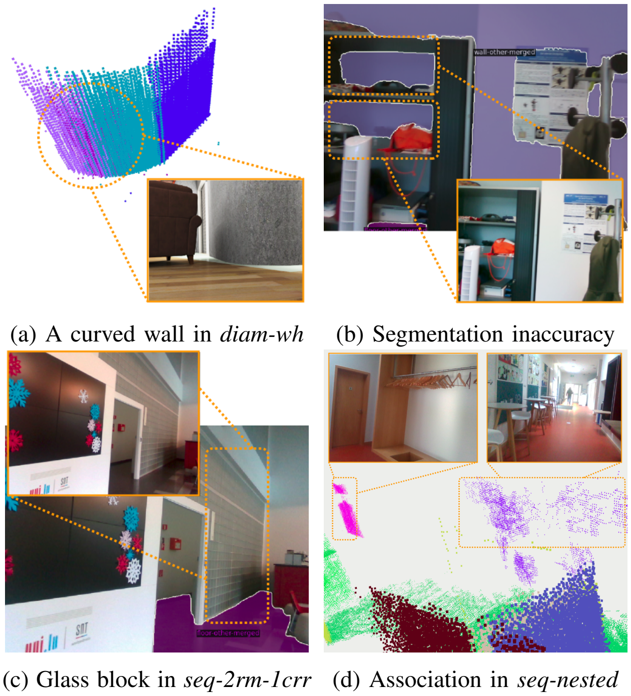
TABLE III: The accuracy of the proposed method in correctly detecting building components.

		Ground truth	**Methodologies**		
Dataset	**Sequence**	*#Components*	*Precision*	*Recall*	*F1-Score*
ICL [30]	deer-gr	5	0.80	0.80	0.80
	deer-wh	10	1.00	1.00	1.00
	deer-w	10	0.91	1.00	0.95
	deer-r	10	1.00	0.80	0.89
	deer-mavf	5	0.80	0.80	0.80
	diam-gr	6	0.75	1.00	0.86
	diam-wh	7	0.71	0.71	0.71
	diam-mavs	4	1.00	0.50	0.67
	diam-mavf	4	1.00	0.50	0.67
ScanNet [31]	seq-0041-01	5	0.83	1.00	0.91
	seq-0103-01	5	1.00	0.60	0.75
	seq-0200-00	5	1.00	1.00	1.00
	seq-0560-00	5	1.00	1.00	1.00
	seq-0614-01	5	1.00	1.00	1.00
	seq-0626-00	5	0.83	1.00	0.91
Collected (in-house)	seq-1rm	5	0.80	0.80	0.80
	seq-4rm-2crr	20	0.86	0.95	0.90
	seq-nested	9	0.70	0.78	0.74
	seq-1rm-1crr	7	0.86	0.86	0.86
	seq-4crr	18	0.84	0.89	0.86
	seq-2rm-1crr	7	0.75	0.43	0.55
	seq-lab	23	0.79	0.83	0.81
Total		**180**	**0.87**	**0.83**	**0.84**

a balanced metric that covers both measurements as follows:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

According to Table III, the presented method functions satisfactorily in most scenarios, consistently acquiring high precision and recall over various datasets' instances. The good results are attributed to the integration of geometric detection followed by semantic validation, improving the accuracy of structural elements in the environment. Slight deviations in performance may occur depending on occlusions, the complexity of the environment, or incomplete data observation in the semantic maps.



(a) A curved wall in *diam-wh* (b) Segmentation inaccuracy

(c) Glass block in *seq-2rm-1crr* (d) Association in *seq-nested*

Fig. 5: Edge-case scenarios posing challenges on the proposed method: **a)** multiple walls detected due to the curved shape of the wall, **b)** misclassification of some parts of the cupboard as a wall, **c)** ignoring glass blocks as walls, and **d)** the same walls are not associated due to pose error.

On the other hand, several factors can be attributed to the causes of inadequate precision or recall results in some cases. A sample is an incorrect association between geometric and semantic entities due to fast camera motion during data acquisition, leading to tracking loss or visual feature mismatching. Additionally, curved wall designs pose challenges on association and validation, as the method mainly relies on planar surfaces for building component detection. Other factors include inaccuracies in semantic segmentation for classifying texture-less flat surfaces (*e.g.,* cupboards and desk drawers) and the existence of walls with glass blocks in the environment. Fig. 5 depicts some of the mentioned limitations of the proposed method in challenging scenarios.

V. Conclusions

This paper presented a real-time pipeline for integrating geometrical planar surface detection and object-level semantic validation to detect building components (in this case, wall and ground surfaces) on RGB-D point cloud data. The pipeline aims to accurately recognize entities critical for high-level scene understanding by applying 3D plane poses extracted from RANSAC-based plane fitting and keeping/discarding those planes after validating their semantic category using a panoptic segmentation. Experimental results revealed that the proposed method could improve scene understanding and representation in 3D scene graphs, besides enhancing the accuracy of a visual SLAM framework by imposing new environment-driven constraints.

As a continuance of the work, we are currently working to incorporate the proposed method into a real-time VSLAM, empowering it to detect high-level structural-semantic entities alongside performing semantic loop closing.

References

[1] D. Hu, V. J. Gan, and C. Yin, "Robot-assisted mobile scanning for automated 3d reconstruction and point cloud semantic segmentation of building interiors," Automation in Construction, vol. 152, p. 104949, 2023.

[2] G. Li, H. Fan, G. Jiang, D. Jiang, Y. Liu, B. Tao, and J. Yun, "Rgbd-slam based on object detection with two-stream yolov4-mobilenetv3 in autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 25, no. 3, pp. 2847–2857, 2023.

[3] H. Pu, J. Luo, G. Wang, T. Huang, and H. Liu, "Visual slam integration with semantic segmentation and deep learning: A review," IEEE Sensors Journal, 2023.

[4] D. Cai, R. Li, Z. Hu, J. Lu, S. Li, and Y. Zhao, "A comprehensive overview of core modules in visual slam framework," Neurocomputing, p. 127760, 2024.

[5] M. N. Favorskaya, "Deep learning for visual slam: the state-of-the-art and future trends," Electronics, vol. 12, no. 9, p. 2006, 2023.

[6] C. Zhang, X. Yang, J. Hou, K. Kitani, W. Cai, and F.-J. Chu, "Egosg: Learning 3d scene graphs from egocentric rgb-d sequences," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 2535–2545.

[7] X. Li, G. Liu, S. Sun, B. Li, and W. Yi, "Rethinking scene representation: A saliency-driven hierarchical multi-scale resampling for rgb-d scene point cloud in robotic applications," Expert Systems with Applications, vol. 243, p. 122881, 2024.

[8] H. Taira, I. Rocco, J. Sedlar, M. Okutomi, J. Sivic, T. Pajdla, T. Sattler, and A. Torii, "Is this the right place? geometric-semantic pose verification for indoor visual localization," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4373–4383.

[9] H. Peng, G. Tong, Z. Li, Y. Wang, and Y. Shao, "3d object detection combining semantic and geometric features from point clouds," Cobot, vol. 1, p. 2, 2022.

[10] C. Chen, Y. Wang, H. Chen, X. Yan, D. Ren, Y. Guo, H. Xie, F. L. Wang, and M. Wei, "Geosegnet: point cloud semantic segmentation via geometric encoder–decoder modeling," The Visual Computer, pp. 1–15, 2023.

[11] H. Shuai and Q. Liu, "Geometry-injected image-based point cloud semantic segmentation," IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1–10, 2023.

[12] Y. Yang, M. Hayat, Z. Jin, C. Ren, and Y. Lei, "Geometry and uncertainty-aware 3d point cloud class-incremental semantic segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 21 759–21 768.

[13] J. Martens, T. Blut, and J. Blankenbach, "Cross domain matching for semantic point cloud segmentation based on image segmentation and geometric reasoning," Advanced Engineering Informatics, vol. 57, p. 102076, 2023.

[14] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Visual slam: What are the current trends and what to expect?" Sensors, vol. 22, no. 23, p. 9297, 2022.

[15] J. Yan, Y. Zheng, J. Yang, L. Mihaylova, W. Yuan, and F. Gu, "Plpf-vslam: An indoor visual slam with adaptive fusion of point-line-plane features," Journal of Field Robotics, vol. 41, no. 1, pp. 50–67, 2024.

[16] H. Yang, J. Yuan, Y. Gao, X. Sun, and X. Zhang, "Uplp-slam: Unified point-line-plane feature fusion for rgb-d visual slam," Information Fusion, vol. 96, pp. 51–65, 2023.

[17] B. Zhang, Y. Dong, Y. Zhao, and X. Qi, "Dynpl-slam: A robust stereo visual slam system for dynamic scenes using points and lines," IEEE Transactions on Intelligent Vehicles, 2024.

[18] J. Dong, M. Lu, Y. Xu, F. Deng, and J. Chen, "Plpd-slam: Point-line-plane-based rgb-d slam for dynamic environments," in 2024 IEEE 18th International Conference on Control & Automation (ICCA). IEEE, 2024, pp. 719–724.

[19] E. Sandström, Y. Li, L. Van Gool, and M. R. Oswald, "Point-slam: Dense neural point cloud-based slam," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 18 433–18 444.

[20] Q. Ji, Z. Zhang, Y. Chen, and E. Zheng, "Drv-slam: An adaptive real-time semantic visual slam based on instance segmentation toward dynamic environments," Ieee Access, vol. 12, pp. 43 827–43 837, 2024.

[21] S. Wei and Z. Li, "An rgb-d slam algorithm based on adaptive semantic segmentation in dynamic environment," Journal of Real-Time Image Processing, vol. 20, no. 5, p. 85, 2023.

[22] P. Cong, J. Li, J. Liu, Y. Xiao, and X. Zhang, "Seg-slam: Dynamic indoor rgb-d visual slam integrating geometric and yolov5-based semantic information," Sensors, vol. 24, no. 7, p. 2102, 2024.

[23] Y. Zhang, Y. Li, and P. Chen, "Tsg-slam: Slam employing tight coupling of instance segmentation and geometric constraints in complex dynamic environments," Sensors, vol. 23, no. 24, p. 9807, 2023.

[24] J. Tang, Z. Feng, P. Liao, L. Chen, and X. Xiao, "Oms-slam: dynamic scene visual slam based on object detection with multiple geometric feature constraints and statistical threshold segmentation," Measurement Science and Technology, vol. 35, no. 10, p. 106307, 2024.

[25] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, R. M. Salinas, and H. Voos, "Marker-based visual slam leveraging hierarchical representations," in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 3461–3467.

[26] A. Tourani, H. Bavle, D. I. Avşar, J. L. Sanchez-Lopez, R. Munoz-Salinas, and H. Voos, "Vision-based situational graphs exploiting fiducial markers for the integration of semantic entities," Robotics, vol. 13, no. 7, p. 106, 2024.

[27] K. G. Derpanis, "Overview of the ransac algorithm," Image Rochester NY, vol. 4, no. 1, pp. 2–3, 2010.

[28] Y. Li, H. Zhao, X. Qi, Y. Chen, L. Qi, L. Wang, Z. Li, J. Sun, and J. Jia, "Fully convolutional networks for panoptic segmentation with point-based supervision," IEEE transactions on pattern analysis and machine intelligence, vol. 45, no. 4, pp. 4552–4568, 2022.

[29] J. Hu, L. Huang, T. Ren, S. Zhang, R. Ji, and L. Cao, "You only segment once: Towards real-time panoptic segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 17 819–17 829.

[30] S. Saeedi, E. D. Carvalho, W. Li, D. Tzoumanikas, S. Leutenegger, P. H. Kelly, and A. J. Davison, "Characterizing visual localization and mapping datasets," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 6699–6705.

[31] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5828–5839.

[32] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, 2021.