# Generation of Indoor Open Street Maps for Robot Navigation from CAD Files

Jiajie Zhang[1], Shenrui Wu[1], Xu Ma[1], Sören Schwertfeger[1]

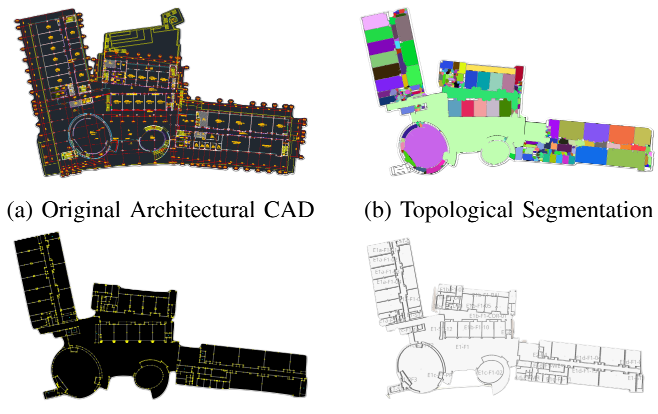
***Abstract*—The deployment of autonomous mobile robots is predicated on the availability of environmental maps, yet conventional generation via SLAM (Simultaneous Localization and Mapping) suffers from significant limitations in time, labor, and robustness, particularly in dynamic, large-scale indoor environments where map obsolescence can lead to critical localization failures. To address these challenges, this paper presents a complete and automated system for converting architectural Computer-Aided Design (CAD) files into a hierarchical topometric OpenStreetMap (OSM) representation, tailored for robust life-long robot navigation. Our core methodology involves a multi-stage pipeline that first isolates key structural layers from the raw CAD data and then employs an AreaGraph-based topological segmentation to partition the building layout into a hierarchical graph of navigable spaces. This process yields a comprehensive and semantically rich map, further enhanced by automatically associating textual labels from the CAD source and cohesively merging multiple building floors into a unified, topologically-correct model. By leveraging the permanent structural information inherent in CAD files, our system circumvents the inefficiencies and fragility of SLAM, offering a practical and scalable solution for deploying robots in complex indoor spaces. The software is encapsulated within an intuitive Graphical User Interface (GUI) to facilitate practical use. The code and dataset are available at https://github.com/jiajiezhang7/osmAG-from-cad.**

***Index Terms*—Mapping, Topometric Map, Robot Navigation**

## I. Introduction

The proliferation of autonomous mobile robots in indoor environments, such as hospitals and warehouses, demands robust and persistent navigation capabilities, for which an accurate environmental map is a fundamental prerequisite. For practical, long-term deployments—often termed "life-long" navigation—maintaining a correct map within a dynamic environment is paramount. Traditional on-site mapping methods, predominantly based on Simultaneous Localization and Mapping (SLAM) [1], exhibit significant drawbacks. The map-building process is often labor-intensive and time-consuming [2], [3]. Moreover, the resulting maps, especially dense metric formats like 2D occupancy grids [4] or 3D point clouds [5], are subject to environmental changes (e.g., rearranged furniture), which can lead to catastrophic localization failures. Their substantial data volume also imposes high storage and computational overhead, rendering them inefficient for long-horizon planning in large-scale settings [6], [7]. Moreover, during data collection for SLAM, often not all rooms are accessible, resulting in incomplete maps.

[1]The authors are with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. {zhangjj2023, wushr2023, maxu2023, soerensch}@shanghaitech.edu.cn



(a) Original Architectural CAD (b) Topological Segmentation

(c) Enhanced OSM in JOSM (d) Rendered in OpenIndoor

Fig. 1: The automatic generation pipeline showcasing key outputs. (a) The original architectural CAD drawing, containing numerous non-essential layers and elements. (b) The result after topological segmentation into an Area Graph, which initially exhibits over-segmented polygons. (c) The refined Area Graph exported to the OpenStreetMap format and visualized in JOSM [8]; this stage includes a post-processing step where small, insignificant areas are pruned or merged. (d) A visualization of our enhanced OSM rendered by OpenIndoor [9], demonstrating that our generated map is fully compatible with standard OSM tools.

To overcome these challenges, we shift the paradigm from on-the-fly mapping to leveraging existing architectural information. This paper presents a mostly automated pipeline that converts standard 2D Computer-Aided Design (CAD) files into an enhanced OpenStreetMap (OSM) [10] representation. The generated map is not merely a geometric blueprint but is enriched with topometric details and a hierarchical structure, encoding both the metric layout and semantic relationships between spaces (e.g., rooms, corridors). This enhanced OSM serves as a robust and persistent foundation for advanced robotic tasks, directly enabling centimeter-level localization [11] and efficient, semantically-aware global path planning [12] and navigation, and thus is a core part of mobile manipulation and embodied AI.

The primary contributions of this paper are therefore as follows:

- We design and implement a complete map generation system that takes a raw architectural CAD floor plan as input and automatically produces an OpenStreetMap

representation incorporating a hierarchical topometric enhancement, termed osmAG [12], which maintains full compatibility with the OSM standard.
- We propose a novel text-to-tag mapping method that systematically interprets semantic text labels within the CAD file (e.g., room names) and translates them into corresponding, standardized OSM tags.
- We develop an automatic merging function that consolidates multiple, single-floor OSM maps of a building into a unified, multi-level map, intelligently identifying and creating the necessary cross-level passages (e.g., stairs, elevators) to ensure topological correctness and cross-level navigation.
- We demonstrate the scalability and real-world applicability of our system by generating and validating a large-scale (approximately 9,000 $m^2$) multi-story map of a complex building, and further validate our approach across a diverse set of real and publicly available CAD floor plans.

## II. Related Works

Integrating prior architectural information is a critical strategy for achieving robust, persistent robot navigation in complex indoor environments, circumventing the limitations of SLAM techniques which struggle with dynamic changes and perceptual ambiguities [13]. Research in this area has evolved from leveraging basic geometric data to incorporating rich semantic and topological information. This section surveys these developments to situate our contribution: an automated pipeline for generating hierarchical topometric OSM maps from CAD files.

### *A. From Geometric to Semantic Architectural Priors*

Early work used geometric primitives from architectural plans to constrain robot localization, for instance, by aligning LiDAR maps to CAD plans [14] or generating point clouds from BIM (Building Information Model) for scan matching [15], [16]. These geometry-centric methods, however, largely ignore semantic context and are fragile in structurally symmetric or feature-sparse environments.

Consequently, research has shifted towards semantic and topological information, constructing graph-based representations that encode architectural meaning and relationships. For example, Shaheer et al. [17], [18] convert building plans into an "Architectural Graph" (A-Graph) to fuse with sensor data for robust localization. Similarly, Karimi et al. [19] use BIM to generate topological maps for semantically-aware planning in ROS, while Zimmerman et al. [20] developed a "high-level semantic map" to aid long-term localization.

A crucial limitation of these advanced methods, however, is their reliance on proprietary or non-standard data formats. The resulting custom graphs and maps hinder interoperability and broader adoption, creating closed ecosystems. Furthermore, many such systems are validated primarily in sparse, construction-like settings and would likely fail in typical furnished environments. In contrast, our approach overcomes these challenges. The map we generate is based on the universal OSM standard and, as shown in previous work [11], supports robust localization even in cluttered, real-world indoor spaces.

### *B. Our Contribution in Context: Standardized Indoor Maps via OSM*

While OpenStreetMap (OSM) is a cornerstone for outdoor robot navigation [21]–[24], its potential for indoor environments remains largely untapped due to the significant manual effort required for map creation. Our work directly addresses this gap by fully automating the generation of the hierarchical topometric map, *osmAG*, which we previously conceptualized in [12], thus transforming a powerful concept into a practical solution.

Distinct from prior art that often produces proprietary graph structures from data-rich BIM or IFC (Industry Foundation Classes) files [18], [19], our system establishes a direct pipeline from a common data source (CAD) to a recognized standard (OSM). The output is not a mere geometric replica but a native hierarchical topometric representation, as defined in [12]. It synergistically integrates metric accuracy, topological connectivity, and a multilevel semantic hierarchy, a much more comprehensive model than purely geometric maps [13] or abstract semantic graphs [20]. Crucially, by adhering strictly to the OSM standard, our method yields a representation that provides a robust foundation for life-long navigation, readily maintained and extended using the vast ecosystem of OSM tools and overcoming the challenge of proprietary, hard-to-maintain formats.

## III. System Methodology

The proposed system systematically transforms architectural CAD floor plans into hierarchical topometric OpenStreetMap (OSM) maps tailored for robotic navigation. An overview of our multi-stage pipeline is depicted in Fig. 2. The process begins with CAD data preprocessing to generate a clean geometric representation, followed by topological segmentation using an AreaGraph [25] structure. This graph then undergoes geometric and logical refinement before being serialized into an enhanced OSM format. Finally, semantic information from the CAD source is attributed to the map, and individual floors are fused into a cohesive multi-story model.

### *A. CAD Preprocessing and Occupancy Grid Generation*

The initial stage of our pipeline converts the raw architectural CAD floor plan into a standardized binary occupancy grid, which serves as the foundation for all subsequent analyses. This conversion involves two primary steps. First, we perform geometry extraction by filtering layers to isolate permanent structural features essential for navigation, such as walls, columns, and windows. To automate this process, we employ a keyword-based filtering mechanism that targets layer names conforming to established architectural standards (e.g., "A-WALL," "A-STAIR"), such as the US National CAD Standard (NCS) [26] and ISO 13567 [27]. For files that deviate

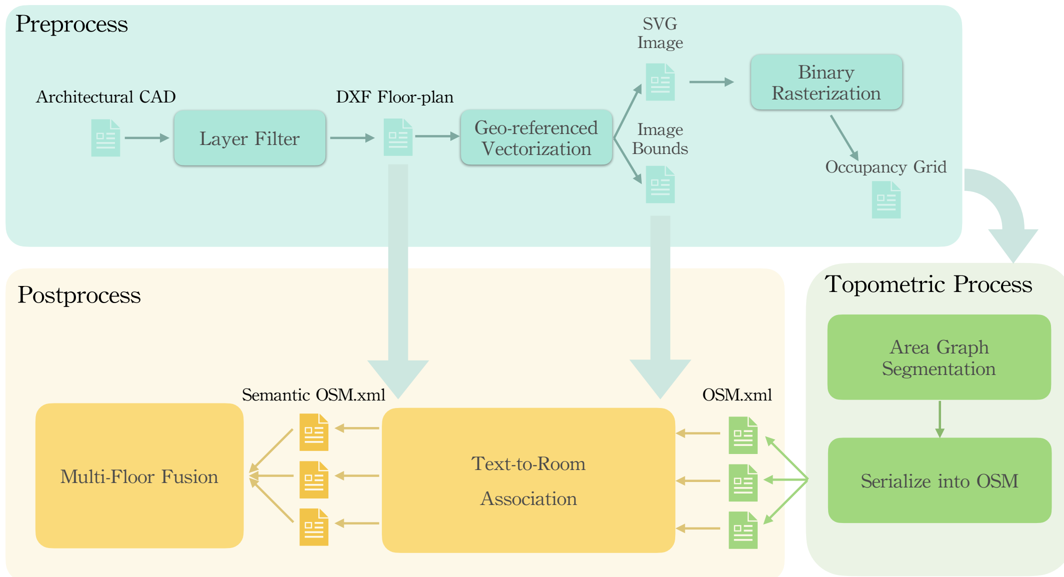



Fig. 2: An overview of the proposed map generation pipeline. The system takes a raw CAD floor plan as input and sequentially processes it through stages of preprocessing, topological segmentation, refinement, serialization, and semantic attribution, culminating in a multi-story, enhanced OSM map ready for robotic navigation.

from these conventions, minimal manual layer selection is required prior to processing.

Naively, one would attempt to then use the vector graphics information in the CAD to directly extract the area ploygons for osmAG. But often, rooms and areas in CAD are not watertight, i.e. the lines for a room drawn by the architect do not always from closed polygons. Thus, in the second step, the filtered vector geometry is rasterized into a high-resolution binary image at a precise metric scale. During rasterization, we apply a morphological processing algorithm to thicken wall structures and bridge minor discontinuities. This enhancement ensures the floor plan's free space is partitioned into well-defined, enclosed connected components, producing a clean occupancy grid. This grid, with unambiguously delineated structures, is a vital prerequisite for robust topological segmentation by the subsequent AreaGraph algorithm.

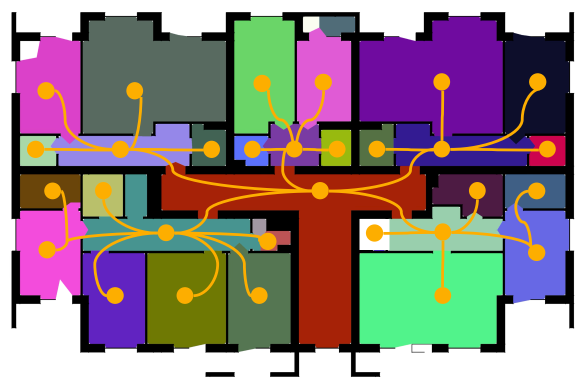


Fig. 3: Illustration of the AreaGraph topological structure derived from a floor plan. Nodes correspond to segmented polygonal areas (e.g., rooms, corridors), and edges represent the passages connecting them, forming the foundational topological map for navigation.

### *B. Topological Graph Generation and OSM Transformation*

The conversion of the raw architectural layout into a structured, navigable map is a two-phase process. First, we generate an abstract topological graph from the floor plan; second, we transform this abstract graph into a geometrically refined and semantically rich OpenStreetMap (OSM) representation.

Our methodology is founded upon the topometric segmentation of the architectural floor plan, adapting the *Area-Graph* representation introduced by Hou et al. [25], [28]. The generation process commences with the construction of a Voronoi Diagram (VD) from the building's geometry, which is systematically pruned to derive a topological skeleton. To prevent the over-segmentation frequently encountered in large, open spaces, we then employ an $\alpha$-shape algorithm to merge adjacent Voronoi cells into semantically coherent polygonal areas, representing physical spaces such as individual rooms and corridors. A comprehensive description of this generation process is elaborated in [25]; here we provide a high-level overview. The result of this phase is a topometric map, the AreaGraph, where nodes are these precisely defined polygonal areas. The graph's edges represent passages that connect adjacent areas, defined geometrically as the shared line segments of their respective polygons. This foundational 2D graph provides not only the connectivity but also the metric information essential for navigation, and serves as the primary layer of our hierarchical map structure.

**Algorithm 1** AreaGraph Generation and OSM Export

**Require:** Occupancy grid $G$, parameters $\alpha$, $\epsilon_{simplify}$, $\theta_{spike}$
**Ensure:** Enhanced OSM map $M_{OSM}$
**Phase 1: AreaGraph Generation**
$V \leftarrow$ COMPUTEVORONOIDIAGRAM($G$)
$T \leftarrow$ EXTRACTTOPOLOGICALSKELETON($V$)
$R \leftarrow$ ALPHASHAPESEGMENTATION($T$, $\alpha$)
$AG \leftarrow$ BUILDAREAGRAPH($R$)
**Phase 2: Geometric and Logical Refinement**
$AG.rooms \leftarrow$ REMOVEDUPLICATEPOLYGONS($AG.rooms$)
$AG.rooms \leftarrow$ MERGESMALLROOMS($AG.rooms$, $A_{min}$, $d_{max}$)
$P_{preserve} \leftarrow$ COLLECTPASSAGEENDPOINTS($AG.passages$)
**for** each room $r \in AG.rooms$ **do**
    $r.polygon \leftarrow$ SIMPLIFYPOLYGON($r.polygon, \epsilon_{simplify}, P_{preserve}$)
    $r.polygon \leftarrow$ REMOVESPIKES($r.polygon$, $\theta_{spike}$, $P_{preserve}$)
**end for**
**Phase 3: OSM Serialization**
$nodes \leftarrow \emptyset$, $ways \leftarrow \emptyset$
$id_{counter} \leftarrow -1$
**for** each room $r \in AG.rooms$ **do**
    **for** each vertex $v \in r.polygon$ **do**
        **if** $v \notin nodes$ **then**
            $nodes[v] \leftarrow id_{counter}$, $id_{counter} \leftarrow id_{counter} - 1$
            CREATEOSMNODE($nodes[v]$, CARTESIANTOLATLON($v$))
        **end if**
    **end for**
    $way_r \leftarrow$ CREATEOSMWAY($id_{counter}$, $nodes[r.polygon]$)
    ADDTAGS($way_r$, {"indoor": "room", "osmAG:type": "area"})
    $ways \leftarrow ways \cup \{way_r\}$, $id_{counter} \leftarrow id_{counter} - 1$
**end for**
**for** each passage $p \in AG.passages$ **do**
    $(v_1, v_2) \leftarrow$ COMPUTEPASSAGEENDPOINTS($p$)
    $way_p \leftarrow$ CREATEOSMWAY($id_{counter}$, $\{nodes[v_1], nodes[v_2]\}$)
    ADDTAGS($way_p$, {"osmAG:type": "passage", "indoor": "door"})
    $ways \leftarrow ways \cup \{way_p\}$, $id_{counter} \leftarrow id_{counter} - 1$
**end for**
$M_{OSM} \leftarrow$ SERIALIZETOOSMXML($nodes$, $ways$)
**return** $M_{OSM}$

While the initial AreaGraph provides the essential topological connectivity, our work significantly extends this concept to create a high-fidelity, persistent map suitable for real-world robot navigation. The original methodology proposed by Hou et al. [25], [28], though foundational for topometric segmentation, was primarily designed for grid maps generated via SLAM and produced a conceptual graph for visualization. This approach could neither process the geometric complexity of raw architectural CAD files directly nor serialize the resulting topometric map into a persistent, machine-readable format for downstream robotic tasks.

To bridge this critical gap, we introduce a robust transformation and serialization pipeline. The process commences with multi-stage refinement of the raw AreaGraph. Due to the rasterization and the subsequent pruning of the VD of this grid map, Area Graphs have polygons with points for each cell. So, first, boundary polygons undergo geometric refinement using an adaptive Douglas-Peucker simplification algorithm [29], which dynamically adjusts its tolerance to preserve curvilinear features common in modern architecture, coupled with an iterative spike-removal procedure to eliminate polygonal artifacts. Critically, vertices corresponding to passage locations are explicitly preserved throughout this process to maintain the topological integrity of the map. E.g., in the example introduced in Fig. 5 the number of osm nodes (polygon points) was reduced from 228,408 to 19,710, a reduction of 91.37%.

Following the geometric cleanup, we perform logical structure refinement to enhance the map's clarity and utility. This step involves identifying and consolidating any geometrically duplicate rooms and merging inconsequently small areas (e.g., tiny slivers from the segmentation process) into their larger, adjacent neighbors.

The final phase of our methodology serializes the refined topological graph into the OpenStreetMap (OSM) XML format, a process distinguished by its precise geo-referencing capabilities. By defining the WGS84 coordinates (longitude and latitude) for a designated origin point alongside the map's metric resolution, the system generates an OSM file where all geometric elements are situated within a global reference frame. This transformation yields a spatially accurate, interoperable, and permanent map artifact that can be seamlessly integrated with the global OSM dataset. Within this structure, each navigable space is encoded as a closed OSM `way` tagged as `indoor=room`. Passages connecting these spaces are likewise represented as `way` elements, but are assigned the custom tag `osmAG:type=passage` and include attributes identifying the two areas they link. This tagging scheme ensures full compatibility with the official OSM standard while embedding the enhanced topological information essential for advanced mobile robot navigation, as demonstrated in previous work [12].

### C. Semantic Text-to-Room Association

To enrich the map with semantic labels, we developed a robust pipeline to associate textual annotations from the CAD file with their corresponding room polygons in the OSM structure. The process begins by extracting text entities and their spatial coordinates from the relevant CAD layers. These coordinates are then transformed into the same reference frame as the room polygons.

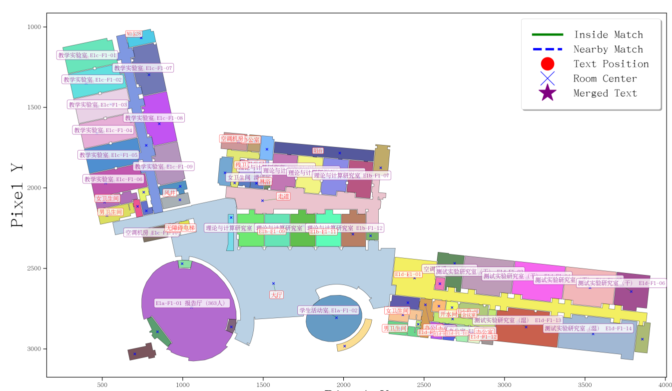



Fig. 4: Illustration of the text-to-room association. Room polygons are shown with colored boundaries. The visualization demonstrates the spatial relationship between extracted text annotations from architectural drawings and their corresponding room polygons in the segmented indoor map.

The core of this stage is a score-based matching algorithm that assigns each text label to the most appropriate room. To accommodate common annotation practices, our scoring function is designed to handle two distinct cases: (1) the text is located inside the room polygon, and (2) the text is located outside but in close proximity to the room boundary. The score is meticulously calculated based on geometric properties, prioritizing labels that are centrally located within a polygon while also correctly associating labels placed just outside their intended room. The text label from the highest-scoring text-room pair is then injected as the `name` tag for the corresponding room in the final OSM data.

The scoring function is formally defined as follows. Let $p$ be the coordinate of a text annotation and $P$ be a candidate room polygon. The assignment score is calculated based on the following definitions:

- $c$: The centroid (center of mass) of the room polygon $P$.
- $A$: The area of the polygon $P$.
- $S = \sqrt{A/\pi}$: The characteristic radius of the room, modeling it as an equivalent circle.
- $d_c = \|p - c\|_2$: The Euclidean distance from the text's position $p$ to the room's centroid $c$.
- $d_b = \mathrm{dist}(p, \partial P)$: The minimum Euclidean distance from the text's position $p$ to the boundary of the room polygon $\partial P$.
- $\rho = d_c/S$: The normalized distance of the text from the centroid relative to the room's characteristic radius.

The matching score is determined by two conditions, governed by the predefined hyper-parameters $\rho_{\max}$ (default 0.7) for the maximum relative center distance and $D_{\max}$ (default 50 pixels) for the nearby matching distance threshold.

**Case 1: Inside Matching** ($p \in P$)
If the text annotation is located inside the polygon, the score is calculated based on its centrality. A higher score is given to text closer to the centroid.

$$\text{score}_{\text{inside}}(p, P) = \begin{cases} 100 - 50 \cdot \rho, & \text{if } \rho \leq \rho_{\max} \\ 50 - 25 \cdot (\rho - \rho_{\max}), & \text{if } \rho > \rho_{\max} \end{cases} \tag{1}$$

**Case 2: Nearby Matching** ($p \notin P \wedge d_b < D_{\max}$)
If the text is outside the polygon but within the distance threshold $D_{\max}$, the score is computed as a weighted sum of a base score, a size factor, and a distance factor. This allows for matching labels placed adjacent to smaller rooms.

$$\text{score}_{\text{nearby}}(p, P) = 40 + 30 \cdot f_{\text{size}} + 20 \cdot f_{\text{dist}} \tag{2}$$

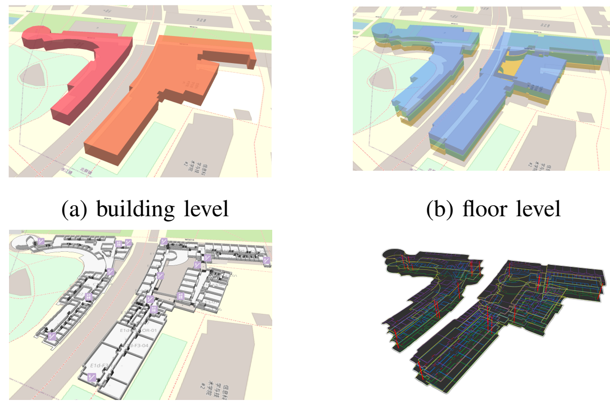
where the contributing factors are:

$$f_{\text{size}} = \frac{1}{1 + \log_{10}(1 + A/10000)} \tag{3}$$

$$f_{\text{dist}} = 1 - \frac{d_b}{D_{\max}} \tag{4}$$

If neither of these conditions is met for a given text-polygon pair, the score is considered zero, and the pair is not a candidate for matching.

**Matching Strategy**: For each text entity, the algorithm computes a score against all room polygons. The room that yields the highest non-zero score is selected as the definitive match for that text label.



(a) building level (b) floor level

(c) room level (d) Rviz visualization

Fig. 5: Results of Hierarchical Multi-Floor Map Fusion. Figures a,b,c depict three distinct hierarchical levels of a ShanghaiTech building as rendered in the standard OpenIndoor viewer [9]. Figure d showcases an enhanced OSM map visualized in Rviz after being processed by osmAG parser [12]. Within this visualization, the red line segments represent vertical passages (e.g. elevators and stairs) that topologically connect adjacent floors. This enhanced representation is specifically designed to enable cross-floor mobile robot localization [11] and path planning [12], and its effectiveness for both applications has been validated, as demonstrated in the accompanying video.

### *D. Hierarchical Multi-Floor Map Fusion*

The culminating step of the topometric processing pipeline involves serializing the refined topological graph into the OpenStreetMap (OSM) XML format, a process distinguished by its precise geo-referencing capabilities. By defining the WGS84 coordinates (longitude and latitude) for a designated origin point alongside the map's metric resolution, the system generates an OSM file where all geometric elements are situated within a global reference frame. This transformation yields a spatially accurate, interoperable, and permanent map artifact that can be seamlessly integrated with the global OSM dataset. Within this structure, each navigable space is encoded as a closed OSM `way` tagged as `indoor=room`. Passages connecting these spaces are likewise represented as `way` elements, but are assigned the custom tag `osmAG:type=passage` and include attributes identifying the two areas they link. This tagging scheme ensures full compatibility with the official OSM standard while embedding the enhanced topological information essential for advanced mobile robot navigation.

Furthermore, this hierarchical paradigm extends to the organization within a single floor. For instance, as illustrated in Fig. 6, a floor can be partitioned into distinct sub-regions, such as different functional sectors. This layered organization is not mandatory but offers a flexible way to structure complex spaces and enhance hierarchical path planning. Users can implement custom hierarchies simply by assigning an

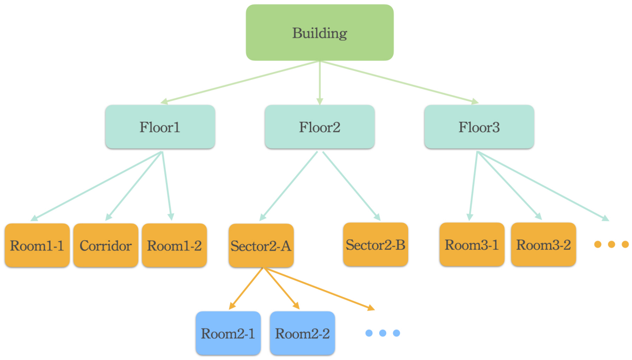


Fig. 6: Illustration of a typical Hierarchical structure of one whole building.

`osmAG:parent` tag to an area, thereby linking it to a larger parent region. The detailed methodology for creating these diverse, user-defined hierarchies is further elaborated in previous work [12].

## IV. EXPERIMENTS

To validate the effectiveness, robustness, and scalability of our proposed automated map generation system, we conducted a series of experiments on a diverse collection of architectural CAD floor plans. As our work presents a novel end-to-end pipeline for converting CAD files into hierarchical topometric OSM maps, there are no existing methods for direct performance comparison. Therefore, our evaluation focuses on quantifying the accuracy and efficiency of our system against manually annotated ground truth.

### A. Datasets

A significant challenge in this research domain is the general scarcity of publicly available, real-world architectural CAD files, as they are often considered proprietary commercial assets. To ensure a comprehensive evaluation, we curated a dataset from two distinct sources:

1) **ShanghaiTech University Campus**: This dataset comprises 6 floor plans from two major buildings on our university campus. Each building has 3 floors, covering a total area of approximately 9000 $m^2$. This large-scale, multi-story dataset serves to demonstrate the system's capability in handling complex, real-world environments and validate our automatic multi-floor fusion functionality.
2) **Public Architectural Repository**: To assess the generalizability of our method, we collected 18 additional floor plans from the public architectural resource ArchWeb.it [31]. This collection is intentionally diverse, encompassing various building types such as schools, hotels, offices, museums, and residential apartments. These plans feature a wide range of structural complexities, room layouts, and annotation styles, providing a robust testbed for the versatility of our system.

### B. Experimental Setup and Metrics

Our evaluation is designed to assess the system's performance across its core functionalities: topological segmentation, semantic association, and overall processing efficiency.

**Ground Truth (GT)**: For each of the 24 CAD floor plans, we manually annotated the ground truth by counting the number of distinct, navigable spaces (rooms and corridors) and the passages (doors or openings) connecting them. This manually curated data serves as the benchmark for our quantitative analysis.

**Quantitative Metrics**: We evaluate the accuracy of the topological structure generation using the standard classification metrics of **Precision**, **Recall**, and **F1-Score**. We also measure the **Semantic Accuracy** of our text-to-room association module, calculated as the percentage of rooms with textual labels in the CAD that were correctly assigned in the final OSM map. Finally, we report the **Average Processing Time** per floor plan to demonstrate the system's efficiency.

The evaluation of our topological segmentation algorithm hinges on a rigorous comparison against manually annotated ground truth (GT). For each element type—namely rooms and passages—we quantify the performance using standard classification metrics derived from the counts of True Positives (TP), False Positives (FP), and False Negatives (FN).

A **True Positive (TP)** is recorded for each room or passage generated by our algorithm that correctly corresponds to a distinct, navigable space or connection in the ground truth. A **False Positive (FP)** occurs when the algorithm generates a room or passage that does not exist in the ground truth; this often results from the over-segmentation of a single large space. Conversely, a **False Negative (FN)** is counted when a ground-truth element is missed entirely by the algorithm, for instance, when two distinct rooms are incorrectly merged into a single polygon (under-segmentation).

Based on these definitions, we processed all 24 CAD files through our automated pipeline. The quantitative results for room and passage segmentation are summarized in Table I.

TABLE I: Quantitative Evaluation of 24 OSM Maps

| Element Type | GT | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Rooms | 612 | 75.73% | 91.71% | 82.88% |
| Passages | 647 | 77.82% | 90.56% | 83.70% |

The results demonstrate a high degree of accuracy for both room and passage identification, with F1-scores of 82.88% and 83.70%, respectively. This indicates that our AreaGraph-based segmentation and refinement pipeline is highly effective at interpreting the structural layout of architectural plans. The primary sources of error (FPs and FNs) typically stemmed from ambiguous geometries in the source CAD files, such as rooms defined by non-enclosing dashed lines or passages that were not clearly delineated from walls, leading to occasional over- or under-segmentation.

For semantic enrichment, the text-to-room association module achieved a **Semantic Accuracy of 91.2%** across all

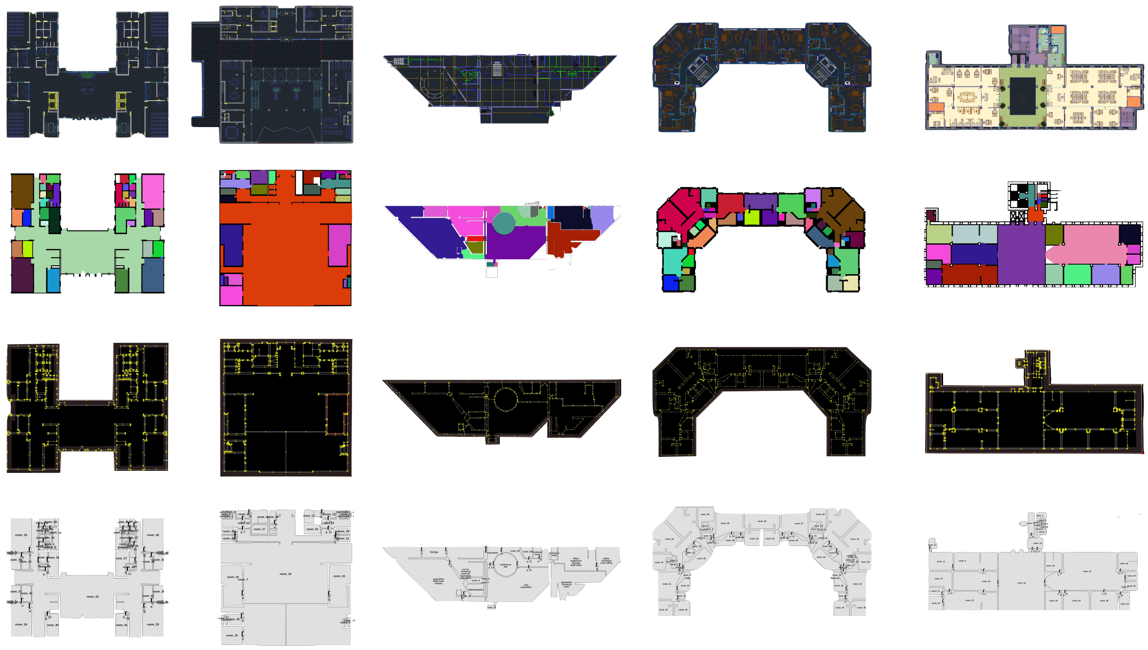


Fig. 7: Qualitative results demonstrating the generalizability of our automated pipeline across 5 diverse architectural styles from the 24 CAD files used here. Each column illustrates the end-to-end process for a single floor plan. From top to bottom: (1) Original CAD input; (2) Intermediate topological *AreaGraph*; (3) Final refined OSM map in JOSM, confirming structural accuracy; and (4) Rendered map in OpenLevelUp [30] with successfully associated semantic labels.

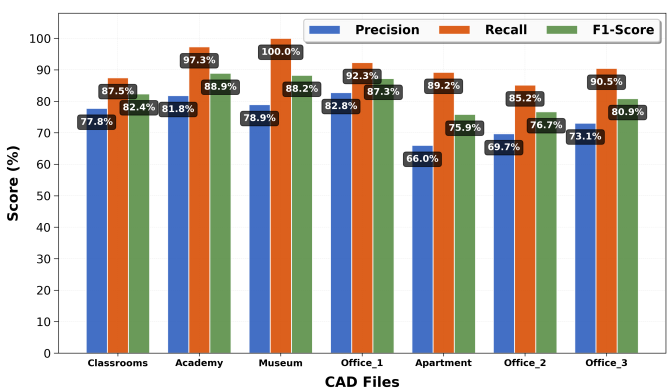



Fig. 8: Performance evaluation of the room segmentation algorithm. The proposed method exhibits a tendency to over-segment complex spaces, resulting in a higher number of false positives. This behavior is intentionally favored to maximize recall, as correctly segmented rooms (True Positives) can be directly utilized, drastically reducing the manual effort of creating a map from scratch. Conversely, the erroneous polygons resulting from over-segmentation can be easily identified and merged with minimal manual intervention during post-processing.

labeled rooms in the dataset. The scoring mechanism proved robust in associating text labels located both inside and in close proximity to their corresponding polygons. Failures were primarily observed in cases of extreme text clutter, instances where a single label was ambiguously placed between two adjacent small rooms, and the presence of extraneous textual annotations in the source CAD file that represented non-label information (e.g., dimensions or material specifications) rather than navigable spaces.

In terms of efficiency, the average processing time for a single-floor CAD plan was approximately 35 seconds on a standard desktop computer (Intel i7, 16GB RAM), confirming the system's practicality for rapid map deployment.

Qualitatively, as illustrated in Figure 7, our system consistently produced high-fidelity OSM maps across the entire dataset. The generated polygonal room structures closely match the original architectural layouts, and the method demonstrates strong generalization across different building types, from the structured grid of an office building to the more irregular shapes found in a museum. Furthermore, the successful generation and fusion of the 6-floor ShanghaiTech University dataset, visualized in Figure 5, confirms the scalability and effectiveness of our hierarchical multi-floor mapping strategy.

## V. Conclusion and Discussion

In this paper, we present an automated system that converts architectural CAD floor plans into a semantically rich hierarchical topometric OpenStreetMap (OSM) representation, addressing a critical bottleneck for the long-term deployment of autonomous robots. Our approach circumvents the labor-intensive nature and fragility of traditional SLAM-based methods by leveraging the permanent structural information inherent in CAD files. The system automates the entire pipeline, from raw data preprocessing to the generation of a topologically coherent, multi-level map, featuring a novel text-to-tag association method and an automatic multi-floor fusion function.

The efficacy and scalability of our system were validated through the successful generation of a map for a large-scale ($9,000\ m^2$) building and across a diverse set of real-world CAD files, achieving high accuracy in both topological segmentation and semantic labeling. By shifting the paradigm from on-site sensing to utilizing pre-existing architectural data, our work provides a practical and efficient solution that enables robust, life-long robot navigation, precise localization, and semantically-aware path planning, as also shown in the video accompanying this paper.

Future work will focus on enhancing the system's autonomy by developing an adaptive parameter selection mechanism for segmentation and extending the algorithm to natively handle layouts with disjointed navigable spaces. Moreover, the foundational principles of our pipeline are generalizable and could be adapted to construct osmAG maps from other modalities, such as large-scale point clouds. These advancements, coupled with potential integration with richer data sources like BIM, will further advance the creation of persistent, life-long maps for robotics.

## REFERENCES


[1] J. Zhang, S. Singh *et al.*, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[2] J. Feng, X. Li, H. Park, J. Liu, and Z. Zhang, "Fr-slam: A slam improvement method based on floor plan registration," in *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2024, pp. 891–896.

[3] L. Mozzarelli, S. Specchia, M. Corno, and S. M. Savaresi, "Automatic navigation map generation for mobile robots in urban environments," *arXiv preprint arXiv:2403.13431*, 2024.

[4] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.

[5] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[6] K. A. Tychola, E. Vrochidou, and G. A. Papakostas, "Deep learning based computer vision under the prism of 3d point clouds: a systematic review," *The Visual Computer*, vol. 40, no. 11, pp. 8287–8329, 2024.

[7] Y. Shi, K. Jiang, J. Li, Z. Qian, J. Wen, M. Yang, K. Wang, and D. Yang, "Grid-centric traffic scenario perception for autonomous driving: A comprehensive review," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[8] OpenStreetMap WiKi contributors, "Josm," https://wiki.openstreetmap.org/wiki/JOSM, 2024, [Online; accessed 17-June-2025].

[9] OpenStreetMap Wiki contributors, "Openindoor," https://wiki.openstreetmap.org/wiki/OpenIndoor, 2024, [Online; accessed 17-June-2025].

[10] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[11] F. Xie and S. Schwertfeger, "Robust lifelong indoor lidar localization using the area graph," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 531–538, 2023.

[12] D. Feng, C. Li, Y. Zhang, C. Yu, and S. Schwertfeger, "Osmag: Hierarchical semantic topometric area graph maps in the osm format for mobile robotics," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–7.

[13] F. Boniardi, A. Valada, R. Mohan, T. Caselitz, and W. Burgard, "Robot localization in floor plans using a room layout edge extraction network," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5291–5297.

[14] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "A pose graph-based localization system for long-term navigation in cad floor plans," *Robotics and Autonomous Systems*, vol. 112, pp. 84–97, 2019.

[15] R. Kaneko, Y. Nakamura, R. Morita, and S. Ito, "Point cloud data map creation from factory design drawing for lidar localization of an autonomous mobile robot," *Artificial Life and Robotics*, vol. 28, no. 2, pp. 314–322, 2023.

[16] H. Yin, J. M. Liew, W. L. Lee, M. Ang Jr, and J. K. Yeoh, "Towards bim-based robot localization: A real-world case study," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 39. IAARC Publications, 2022, pp. 71–77.

[17] M. Shaheer, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Robot localization using situational graphs (s-graphs) and building architectural plans," *Robotics*, vol. 12, no. 3, p. 65, 2023.

[18] M. Shaheer, J. A. Millan-Romera, H. Bavle, J. L. Sanchez-Lopez, J. Civera, and H. Voos, "Graph-based global robot localization informing situational graphs with architectural graphs," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9155–9162.

[19] S. Karimi, R. G. Braga, I. Iordanova, and D. St-Onge, "Semantic optimal robot navigation using building information on construction sites," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 38. IAARC Publications, 2021, pp. 57–64.

[20] N. Zimmerman, T. Guadagnino, X. Chen, J. Behley, and C. Stachniss, "Long-term localization using semantic cues in floor plan maps," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 176–183, 2022.

[21] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1645–1650.

[22] M. Á. Muñoz-Bañón, E. Velasco-Sanchez, F. A. Candelas, and F. Torres, "Openstreetmap-based autonomous navigation with lidar naive-valley-path obstacle avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 428–24 438, 2022.

[23] C. A. B. Przewodowski Filho, F. S. Osório, and V. Grassi Júnior, "Global localization using openstreetmap and elevation offsets," *Journal of the Brazilian Computer Society*, vol. 30, no. 1, pp. 264–273, 2024.

[24] O. Bashkanov, M. Seidel, M. Yakymets, N. Daupayev, Y. Sharonov, T. Assmann, S. Schmidt, and S. Zug, "Exploiting openstreetmap-data for outdoor robotic applications," in *2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. IEEE, 2019, pp. 1–7.

[25] J. Hou, Y. Yuan, and S. Schwertfeger, "Area graph: Generation of topological maps using the voronoi diagram," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 509–515.

[26] S. C. Spangler, R. Fujan, G. Piotrowski, and B. Baker, "A/e/c graphics standard," 2023.

[27] B.-C. Björk, K. Löwnertz, and A. Kiviniemi, "Iso 13567-the proposed international standard for structuring layers in computer aided building design," 1997.

[28] J. Hou, Y. Yuan, Z. He, and S. Schwertfeger, "Matching maps based on the area graph," *Intelligent Service Robotics*, vol. 15, no. 1, pp. 69–94, 2022.

[29] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[30] OpenStreetMap WiKi contributors, "Openlevelup," https://wiki.openstreetmap.org/wiki/OpenLevelUp, 2025, accessed: 2025-06-17.

[31] Archweb, "Archweb - disegni di architettura," https://www.archweb.it/, 2025, accessed: 2025-06-17.