An Automated Pipeline for Generating a Multi-Layered Spatial Map for Indoor Robot Navigation from BIM Data

Festus Basimtaal AYEMBILLA[1], Taeyoung KIM[1], Min-Koo KIM[1], JoonOh Seo[2], Jung In KIM[3]

[1]Department of Architectural Engineering, Faculty of Engineering, Chungbuk National University, Republic of Korea
[2]Department of Architectural Engineering, Hanyang University, Seoul, the Republic of Korea
[3]School of Civil and Environmental Engineering, Kookmin University, 02707, Korea
festusayembilla@chungbuk.ac.kr, kimty99@chungbuk.ac.kr, joekim@chungbuk.ac.kr, junoseo@hanyang.ac.kr, jikim07@kookmin.ac.kr

Abstract

Accurate and machine-readable indoor maps are essential for autonomous systems, such as mobile robots, to operate effectively. Existing traditional mapping methods, such as manual piloting or Simultaneous Localization and Mapping (SLAM), are often time-consuming, lack semantic details, and are affected by environmental changes. Building Information Modeling (BIM) has recently been utilized to provide detailed building data, but it is limited in its direct deployment for robotics applications. To address these limitations, this paper proposes a fully automated pipeline and a standardized, multi-layer map data model that converts Building Information Models (BIM), specifically in the Industry Foundation Classes (IFC) format, into robot-usable navigation maps. The proposed technique generates an integrated map model comprising three distinct yet interconnected layers. These layers include (1) a metric occupancy grid, (2) a semantic entity layer, and (3) a topological graph. The generated Spatial Map Model was evaluated in a simulation experiment to verify its application. The findings illustrated the pipeline's robustness in automatically producing an integrated, multi-layer map that improves localization and enables context-aware navigation for robots in an indoor environment.

Keywords –

BIM, IFC, semantic mapping, topological maps, occupancy grid, ROS2

1 Introduction

The use of autonomous mobile robots in service roles, particularly within large and dynamic indoor environments such as hospitals, is a rapidly growing field of research and application. Despite their enhanced efficiency in logistics, sanitation, and patient support tasks, they are not widely adopted in these industrial settings [1], [2]. This is due to the challenge of generating robust and intelligent navigation systems to guide these robots [3]. This limitation is often observed in indoor environments where there is no universal positioning system, and is often characterized by complex, structured layouts [4], [5].

Traditional methods for creating robot maps include manual piloting and Simultaneous Localization and Mapping (SLAM). These techniques are time-consuming and error-prone, especially for large facilities, and are not scalable in environments that undergo frequent reconfiguration. Although SLAM systems can generate maps autonomously, they rely on metric information and lack the semantic understanding required for complex indoor robot tasks. Furthermore, SLAM systems are susceptible to drift and errors in feature-poor or repetitive areas, which are common in complex indoor corridors [6], [7].

Concurrently, the Architecture, Engineering, and Construction (AEC) industry has widely adopted Building Information Modeling (BIM) as a standard for designing and managing buildings. Building Information Models (BIM) offer a rich source of a priori information that can address these shortcomings [8]. BIMs serve as digital twin models and often contain detailed geometric and semantic data about a building's layout, including spaces, connectivity, and object properties. Hence, leveraging this "as-designed" data can eliminate the need for preliminary manual mapping and provide a powerful prior for robotic tasks [9], [10].

While the concept of using BIM for robotics is not new, a significant gap remains. Previous work has explored the conversion of BIM data for navigation, but existing approaches often require significant manual intervention, lack standardization, or fail to produce outputs directly usable by robotic navigation stacks. Additionally, they typically do not address the need for multi-floor navigation or provide semantic enrichment capabilities essential for context-aware robot behavior. This paper addresses these challenges by presenting an automated IFC-to-Spatial Map pipeline that produces a multi-layer spatial representation optimized for robotic navigation. This paper presents three key contributions, including, (1) A Multi-Layer Spatial Map Model. The paper proposes a multi-layer indoor map model derived from IFC that comprises (a) an entity layer, (b) a topological network, and (c) occupancy grids, all aligned to a unified coordinate system (UCS). (2) an automated IFC-to-map pipeline that generates robot-usable navigation outputs for ROS2; and (3) a simulation-based validation demonstrating the feasibility of same-floor, cross-floor, and semantic-goal navigation.

The remainder of this paper is organized as follows: Section 2 reviews related work across relevant domains. Section 3 provides an overview of our system architecture. Section 4 details the multi-layer spatial map model, while Section 5 validates the generated Spatial Map Model. Section 6 concludes and discusses future work.

2 Related Work

2.1 Metric and Semantic Mapping for Robots

Classic robot mapping techniques such as occupancy grid mapping and graph-based SLAM produce 2D or 3D metric maps for localization and path planning. These maps represent free and occupied space but lack semantics for rooms, doors, or functional areas [7], [11]. On the other hand, semantic SLAM augments geometric mapping with object recognition and labeling, typically identifying individual objects rather than building-scale spaces [12]. But, robot navigation systems built on metric maps cannot issue goal commands based on semantic details or building logic [13]. Hence, robot navigation systems will require additional semantic understanding to be effective and efficient in carrying out complex indoor tasks in industrial environments.

2.2 BIM/IFC for Indoor Navigation

BIM-based indoor navigation has received increasing attention, particularly in pedestrian wayfinding and facility management applications [14], [15]. For instance, Liu et al. [14] proposed a Building Information Modeling-based Positioning and Navigation (BIMPN) framework, which combines an entity layer with a topological network layer to support route planning and spatial queries. The proposed workflow extracts floor-plan information from BIM models, generates corridor centerlines, and defines Step Nodes to construct a navigation network. The authors evaluated the proposed technique through scenario-based navigation tasks using a mobile positioning system. Although BIMPN primarily addressed human navigation requirements and web-based visualization platforms, the findings demonstrated that BIM-derived semantic navigation systems can provide high-level routing information compared to purely metric-based map representations [16], [17].

Related studies have explored the use of BIM models to generate waypoints and mission plans in construction robotics [18], [19], [20]. Braga et al. [16], [18] developed the BIRS system, which constructs a topological hypergraph and occupancy grid from IFC for construction-site robots, but generates these as separate artifacts without coordinate-level integration. Follini et al. [21] extract navigation maps, semantic JSON files, and waypoints from IFC as individual outputs requiring manual assembly. A recent framework [22] combines BIM-derived occupancy grids with a semantic mapping dictionary for localization, but omits a topological network layer. Zhu et al. [17] propose IFC-Graph for semantics-based indoor connectivity, producing topology without metric grids and targeting human pathfinding. These approaches produce individual spatial representations that require additional steps to assemble into a robot-deployable system. This gap motivates an automated pipeline that generates a coordinated, multi-layer map with unified coordinates, directly consumable by standard robotic navigation stacks [18].

2.3 Indoor Spatial Standards and Interoperability

Several standards support indoor spatial modeling and cross-domain interoperability. Industry Foundation Classes (IFC) is the dominant open BIM standard which provides geometry, semantic properties, and spatial relationships that can serve as a rich source for automated map generation workflows [23]. IndoorGML was developed specifically to represent indoor spatial topology and navigation networks, and ongoing extensions (IndoorGML 2.0) broaden its practical applicability across multiple data formats [24]. CityGML further supports semantic 3D modeling at the city scale and can be extended to capture indoor information through domain-specific extensions, enabling integrated urban-building workflows [25].

To support the conversion and reuse of these standards, prior work has demonstrated pathways that translate IFC entities into graph-based or dual-space representations, enabling the construction of navigable nodes and edges while preserving semantic attributes [23], [26]. For industrial deployment, JSON-LD provides a practical, web-based format for exporting semantic entities and relationships in a way that is compatible with modern APIs and linked data workflows [27]. JSON-LD's use in smart building knowledge graphs and digital twin systems suggests that it can support structured semantics and cross-domain integration when combined with appropriate contexts [28] [29]. These considerations motivate the inclusion of JSON-LD as an optional interoperability output alongside domain-specific robot artifacts.

3 Pipeline Overview and Spatial Map Model

This section describes the proposed pipeline and the resulting spatial map representations. As shown in Figure 1, the pipeline takes as input (i) a BIM model in IFC format and (ii) a set of configuration settings that control spatial standardization and map generation. The configuration settings specify the parameters needed to operationalize the conversion, including UCS specification, grid-map parameters, and navigation graph settings. The pipeline first performs input validation and computes a UCS transformation, which is stored and reused when generating each map layer. This ensures that the entity layer, navigation network, and occupancy grids are produced in a consistent coordinate frame, which is essential for an integrated multi-layer mapping system and for exporting robot-ready artifacts. Finally, the pipeline produces a unified multi-layer spatial map and associated exports, including robot-consumable packages and (optionally) a semantic interoperability representation.

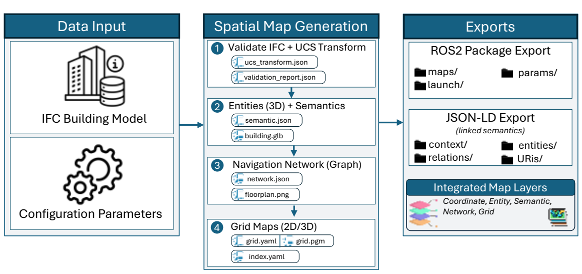

Figure 1. Overview of the proposed IFC-to-spatial-map pipeline.

3.1 Unified Coordinate System (UCS)

IFC geometry is often defined through hierarchical local placements rather than a single global reference frame, and consistent georeferencing metadata may be absent or inconsistently represented across authoring tools [30]. This can lead to coordinate discrepancies that complicate downstream integration, including BIM–GIS alignment and robotic map deployment [31]. To address this challenge, a Unified Coordinate System (UCS) was defined and applied to align the generated maps with the IFC coordinates. In implementation, the placement hierarchy was traversed, nested transforms were composed, and explicit map conversion metadata was present to transform all relevant geometries into a consistent origin, axis conversion, and unit system. Each building storey was identified using IfcBuildingStorey, and each storey was assigned a vertical band using its elevation information while sharing a common horizontal coordinate frame to support cross-floor alignment. When global coordinates are unavailable, a fallback UCS is defined based on the building's bounding box to ensure that all downstream outputs remain spatially consistent for map generation and export. The pipeline also includes checks to handle imperfect IFC inputs. First, it validates that required spatial hierarchy elements (IfcProject, IfcBuilding, IfcBuildingStorey, IfcSpace) are present, reporting any missing elements as warnings instead of stopping the process. For unit-scale detection, it uses a three-step fallback including (1) SI units, (2) a conversion-based lookup, and (3) a default value of 1.0. When assigning storeys, it groups actual element elevations within a 0.25 m range to find mismatches with the declared storey heights. If an element is missing geometry, the pipeline skips it without stopping. These methods allowed us to process IFC models created with different authoring conventions, as shown in Section 4.

3.2 Spatial Map Layers

Figure 2 illustrates the proposed multi-layer spatial map model. The representation is organized into three coupled layers: an entity layer, a topological navigation network layer, and an occupancy grid layer, which share a common UCS so that semantic entities, connectivity, and metric-free space remain mutually aligned.

3.2.1 Entity layer

Figure 3 shows the entity representation of the spatial Map. This layer captures semantic building elements relevant to navigation, including rooms (IfcSpace), corridors, doors (IfcDoor), stairs (IfcStair), elevators (IfcTransportElement), walls (IfcWall), and other IfcElements, such as furniture. The building's entities are grouped by storey using the IfcBuildingStorey hierarchy. Walls, columns, and permanent fixtures are projected onto the ground plane and buffered to account for the robot's footprint and safety distances. Rooms, doors, stairs, and elevators are extracted with their names, types, and geometric boundaries. Each entity is assigned a unique identifier and floor index and is represented by its 3D geometry, 2D footprint, and attributes. The layer is exported as a JSON file and a glTF/GLB mesh for visualization and software application deployment.

3.2.2 Topological network layer

The topological network layer is constructed as a direction graph whose nodes represent navigable spaces and connectors (rooms, corridor segments, doors, stairs, elevators). Edges represent adjacency relations (room–door, corridor–door) and traversal options. They are added between nodes that share a common opening or boundary. Corridors are identified, and a centre line is sampled into step nodes to improve navigation within long, elongated spaces. Step nodes subdivide long corridors to improve path granularity. The network generates connectivity across floors via elevator or stair nodes. This layer enables high-level route planning and semantic queries such as *"shortest path from Room 103 to Lift 1"*. The resulting graph is stored as adjacency lists and coordinates.

3.2.3 Occupancy grid layer

For each floor, a 2D and 3D occupancy grid is generated at a chosen resolution. Free cells correspond to walkable areas, while occupied cells correspond to walls and obstacles. Doors and connectors are carved open to allow passage. The grid is exported as a PNG/PGM image and a 3D GLB model, with a YAML metadata file specifying the resolution, origin, and free/occupied thresholds. For ROS2 Nav2, the YAML file includes the map origin, resolution, and thresholds.

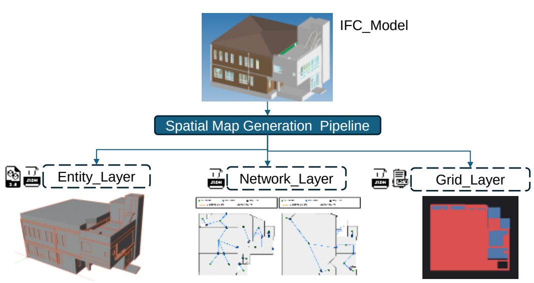

Figure 2. Multi-Layer Spatial Map Generated from the IFC model input.

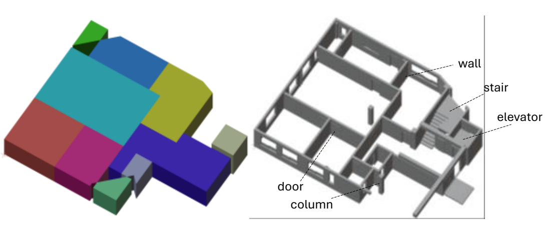

Figure 3. Entity Model. (a) Room geometry boundaries obtained from IfcSpaces. (b) 3D building model and spatial representations.

3.3 Outputs and Interoperability

The pipeline exports four categories of files: (1) entities.json and building.glb for the entity layer; (2) topology-graph.json for the network layer; (3) map.pgm and map.yaml for the 2D occupancy grids; (4) occupancy_3d.npz and occupancy_3d.json for the 3D voxel grids, with optional occupancy_floor.glb for 3D visualization. The export stage converts IFC-derived spatial data into robot-usable outputs through several processing steps. These include composing hierarchical IFC placements into a unified coordinate frame, rasterizing 3D wall and obstacle geometry into floor-based occupancy grids while preserving door openings, generating the PGM/YAML file pairs required by ROS2 Nav2, and resolving coordinate convention differences during GLB and ROS export. These steps are performed automatically by the pipeline and enable direct use of the generated maps in robot navigation workflows. Additionally, we generate a ROS2 Nav2 package that contains grid maps, launch configurations, and multi-robot coordination nodes, enabling immediate navigation experiments with robot fleets. An optional JSON-LD serialization uses the W3C Building Topology Ontology (BOT), OGC GeoSPARQL, and Schema.org vocabularies to express entities and topology, facilitating integration with facility management systems. For example, a robot task planner can resolve a command such as "navigate to Room 101" by querying the JSON-LD graph for the corresponding spatial node and coordinates, which are then passed to the navigation network for route computation. Figure 4 shows an excerpt from the entity layer export.


```
{
 "@context":{
  "bot":"https://w3id.org/bot",
  "geo":"http://www.opengis.net",
  "schema":"https://schema.org/"
 },
 "@type":["bot:Building","spatial_map:EntityLayer"],
 "bot:hasStorey": [{
  "@id": "storey/first-floor",
  "bot:hasSpace": [{
   "@id": "space/room-101",
   "schema:name": "Room 101",
   "geo:hasGeometry": { "geo:asWKT": "POLYGON((...))"}
  }]
 }]
}
```


Figure 4. JSON-LD excerpt of the exported entity layer based on the Building Topology Ontology (BOT).

4 Experiment

4.1 Experiment Data

The proposed pipeline was evaluated on the Phantasy Office Building obtained from the KIT IFC Examples website [32]. The building has five floors: four aboveground and one underground. Figure 4a shows the 3D model of the building in Open IFC Viewer. A single floor view is isolated in Figure 4b.

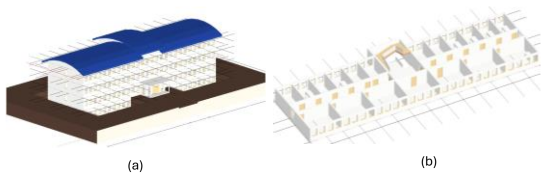

Figure 5. 3D views of the IFC office building model used for evaluating the proposed pipeline.

4.2 Spatial Map Model of the Experimental Data

To represent the extracted 3D building geometry in a portable format, the entity geometry was exported as a GLB file, the binary form of glTF 2.0. This format is widely supported across web-based and desktop visualization tools as well as common 3D engines. Compared with text-based exports (e.g., OBJ), storing geometry and associated appearance information in a single binary container simplifies file handling. During the export stage, the pipeline also resolves coordinate convention differences by converting the IFC model orientation to match the target glTF convention. This enables consistent rendering without manual axis correction. Each exported mesh is assigned a unique identifier derived from its IFC class and GlobalId, which allows direct linking between the geometry in the GLB file and the corresponding entries in entities.json. This linkage supports programmatic access to both geometry and metadata for tasks such as visualization, semantic querying, and robot-oriented scene interpretation. Using this representation, the pipeline generates three interconnected spatial map layers from the experimental building model.

The entity layer stores relevant building components that were extracted from the IFC model. For each entity, its 3D geometry, 2D projection, storey membership, and selected IFC attribute are reserved. The entity layer supports semantic queries such as retrieving all doors on a specific floor or locating candidate vertical connectors (e.g., staircases or elevators) for inter-floor navigation.

The network layer represents the spatial connectivity as a graph. Following the node categorization used by Liu et al. [14], nodes are classified into three types as highlighted in Table 1. Target Nodes represent the navigation destinations and are placed within room regions (IfcSpace) using an interior-point strategy based on centroid placement. The Connection Nodes represent transitions between adjacent spaces. Step Nodes are inserted as intermediate waypoints along corridors and stair segments to increase path resolution and stabilize route generation. The graph edges generate adjacency relationships, and traversal costs are computed from Euclidean distance, enabling shortest-path planning using standard algorithms such as Dijkstra or A*.

Table 1. Navigation Network Node Classification.

Node	Entity connection
Target Node	Rooms/ifcSpace
Connectivity Node	Doors/Staircase/Elevators
Step Node	Corridors/Staircase

The generated occupancy grid layer generates per-floor 2D occupancy grids. This layer provides metric representations for collision detection and SLAM-compatible localization. Each grid cell (0.05 m resolution) is classified as free (255), occupied (0), or unknown. The grids are exported in PGM format with accompanying YAML metadata for direct use with the ROS2 Nav2 map server.

Table 2 presents the spatial map model statistics for the Phantasy Office Building, organized by floor. The table includes counts of building elements and node distributions across the three node categories. Figures 6 visualize the generated spatial map models for the office building.

Table 2. Spatial Map Model Statistics for the Phantasy Office Building

Floor	Building Entities	Network Nodes			Total Nodes
		Target	Connectivity	Stair	
Basement	63	17	32	14	63
Ground Floor	80	18	36	26	80
Floor 1	92	22	44	26	92
Floor 2	90	22	42	26	90
Attic	15	3	0	12	15
Total	340	82	154	104	340

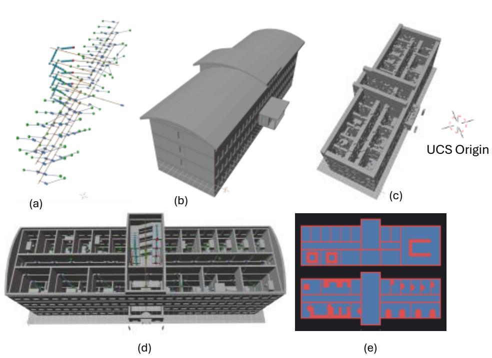

Figure 6. Multi-layer spatial map output. (a) Topological navigation network graph showing

inter-floor connectivity. (b) 3D building entity model (GLB export). (c) 3D building grid model. (d) 3D visualization of Entity + Network Layers. (e) 2D occupancy grids of two floors.

5 Validation and Discussion

The generated spatial maps were validated in ROS2 RViz using a multi-layer visualization and an RViz multi-robot simulation. The validation checks layer alignment (mesh, topology, and occupancy grids), grid integrity for collision checking, and feasibility of same-floor and cross-floor routes.

Evaluation metrics include: (i) map completeness and coverage (navigable area, grid resolution, free/occupied ratios), (ii) topological connectivity (node and edge counts, floor connectivity), (iii) layer alignment (grid, mesh, and graph overlays in RViz), and (iv) navigation performance (success rate, average waypoints, and path length).

5.1 Spatial Map Validation

All five floors passed validation with consistent coordinate alignment across layers. The topological network contains 340 nodes (82 room nodes, 154 door nodes, and 104 stair nodes) with 605 edges that consists of room-door adjacencies, corridor-door connections, stair transitions, and corridor segment bridges. Four vertical stair connections link consecutive floors (Basement↔Ground, Floor↔First, Floor↔Second, Floor↔Attic). Grid resolution of 0.05 m provides sufficient precision for robot footprint clearance. In RViz, each floor occupancy grid (PGM/YAML) is loaded as a separate Map display and overlaid with the building mesh and topological markers, confirming that door openings and corridor free space align with the grid. All layers share the map frame with a static map-to-odom transform, and no systematic offsets were observed.

A sensitivity analysis was conducted on the KIT building model to examine the effect of grid resolution and wall-buffer settings on the generated navigation outputs. As shown in Table 3, increasing the grid size from 0.05 m to 0.10 m reduces the number of cells by 75% while preserving 96.4% of the navigable area. This indicates that a coarser grid can still represent the building effectively while using less memory. The small loss in navigable area (3.6%) occurs mainly because narrow spaces near walls are captured less accurately at the lower resolution.

Table 4 shows that varying the wall buffer between 0.02 m and 0.10 m did not affect the network structure or the navigation results for this model. This is because the connections are validated using space boundaries rather than wall geometry alone. In tighter layouts, a larger buffer may block some connections, but in this case the default value of 0.05 m provided a good balance.

Table 3. Effect of grid resolution on occupancy grid statistics.

Resolution(m)	Navigable Area (m2)	Free Ratio (%)	Total Cells
0.05	2,028	46.0	1,763,460
0.10	1,954	44.3	441,350

Table 4. Effect of wall-buffer size on network connectivity and navigation performance

Wall Buffer	Nodes	Edges	Success Rate	Average Path Length (m)
0.02	630	512	100%	40.43
0.05	630	512	100%	40.43
0.10	630	512	100%	40.43

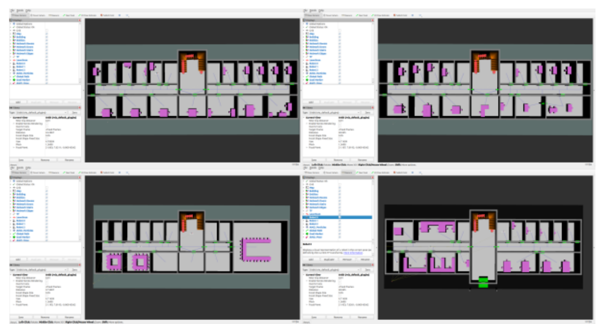

Figure 7. RViz validation snapshots for single-floor views.

5.2 Navigation Path Planning Validation

Navigation path planning was validated in RViz using graph routing (Dijkstra) with occupancy-grid refinement (A*). The evaluation uses a 15-case suite (5 same-floor, 5 cross-floor, 5 semantic queries) implemented in the ROS2 validation script, and Table 6 summarizes the results. Average waypoints and path lengths are computed from the shortest-path graph outputs. The five semantic goal resolution tests evaluate the system's ability to translate user-provided room identifiers into topological graph nodes and compute a valid route. The specific queries are highlighted in Table 5.

Table 5. Semantic Queries used in validation.

Query	Type
Room 002 to Room 006	Same floor
Room 105 to Room 115	Same floor
Room 201 to Room 218	Same floor
Room 101 to Room 315	Cross Floor
Room 008 to Room 301	Cross Floor

Table 6. Navigation Test Results (graph-based planning)

Test Scenario	Success Rate	Avg. Waypoints	Avg. Path Length (m)
Same-Floor Navigation	5/5 (100%)	16.4	23.5
Cross-Floor Navigation	5/5 (100%)	77.2	76.2
Semantic goal resolution	5/5 (100%)	48.4	45.1
Overall	15/15 (100%)	47.3	48.3

Same-floor navigation achieved a 100% success rate across all five test cases, demonstrating that the topological network correctly encodes room-door-corridor connectivity within each floor. Test paths included navigation between adjacent rooms, between rooms separated by corridor segments, and between rooms at opposite ends of the building. For each floor segment, the route is refined on the occupancy grid with A* to prevent wall crossings and keep the path within free cells, which is directly visible in RViz. Cross-floor navigation achieved 100% success across all five test cases, indicating complete vertical connectivity for the evaluated room pairs. The longest cross-floor path (Room 001 to Room 301, spanning four floors) was computed with 126 waypoints, demonstrating the network's capacity for complex multi-floor routing when the vertical links are complete.

The 100% success rate in semantic goal resolution (5/5 tests) demonstrates the utility of preserved IFC attributes for high-level robot task planning and confirms complete vertical connectivity in the evaluated graph. In successful trials, the system mapped user-provided room identifiers (e.g., “Room 002” and “Room 006”) to corresponding topological nodes and computed a route using Dijkstra’s shortest-path algorithm. The same technique also confirmed that the evaluated graph contained valid vertical connectors for cross-floor tests, since goals on different storeys remained reachable through the network.

Overall, the results suggest that standardized BIM/IFC models can provide a practical prior for robot navigation map preparation, substantially reducing reliance on separate manual map-building steps during initial deployment. For existing facilities without BIM, SLAM surveys followed by BIM retrofitting remain necessary. However, for new construction projects with BIM deliverables, the proposed pipeline enables robots to be deployed immediately upon facility handover. The JSON-LD export supports integration with facility management systems (e.g., NGSI-LD context brokers), enabling semantic queries across building operations and robot tasking. This integration pathway addresses a critical gap in the literature, where most prior work treats robot navigation maps as isolated artifacts, whereas facilities increasingly require unified digital twins that serve both human operations and autonomous systems.

Compared with SLAM-based and manual map-generation workflows, the proposed approach offers a different trade-off. It operates directly from BIM without requiring physical site access and preserves semantic information that is typically absent from purely metric maps. SLAM-based methods remain effective for capturing as-is conditions, but they are susceptible to drift in repetitive indoor environments and do not inherently provide persistent room- or object-level semantics [6], [7]. Manual map creation can produce usable navigation assets, but it is labor-intensive and difficult to reproduce consistently across buildings. Existing IFC-to-navigation approaches reduce the need for on-site surveying, yet they generally produce only partial outputs, such as topology, geometry, or occupancy maps in isolation. In contrast, the proposed pipeline automatically generates an integrated, multi-floor, ROS2-ready output consisting of occupancy grids, a topological navigation network, and semantic entity data from a single IFC source.

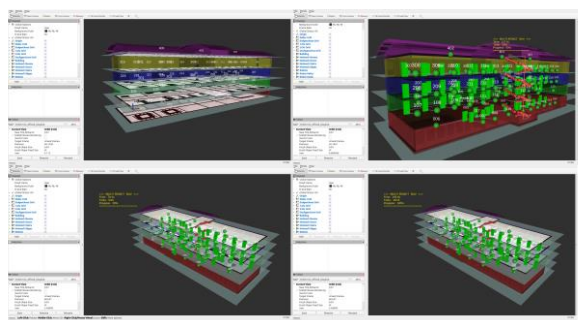

Figure 8. Multi-floor RViz visualizations showing layered occupancy grids and topological network connectivity across floors.

6 Conclusion

This paper introduced an automated pipeline for creating multi-layer spatial maps from IFC building models for ROS2 indoor robot navigation. Evaluation on two structurally distinct buildings demonstrated successful spatial map generation without parameter tuning, together with complete navigation success in simulation tests. The pipeline generates robot-usable map outputs in minutes, compared with the more time-consuming effort typically required for manual or SLAM-based preparation. Limitations remain in the dependence on IFC model quality and the absence of richer semantic layers such as POIs and functional zones, though robustness strategies mitigate several common modeling issues. Future work will focus on semantic enrichment and automated IFC quality assessment and repair as well as generalization test of different IFC models to assess the robustness of this pipeline.

Acknowledgments

This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant Number: RS-2025-02532980).

7 References

[1] G. Fragapane, H.-H. Hvolby, F. Sgarbossa, and J. O. Strandhagen, "Autonomous Mobile Robots in Hospital Logistics," in *Advances in Production Management Systems. The Path to Digital Transformation and Innovation of Production Management Systems*, B. Lalic, V. Majstorovic, U. Marjanovic, G. von Cieminski, and D. Romero, Eds., Cham: Springer International Publishing, 2020, pp. 672–679. doi: 10.1007/978-3-030-57993-7_76.

[2] Md. A. K. Niloy *et al.*, "Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review," *IEEE Access*, vol. 9, pp. 35338–35370, 2021, doi: 10.1109/ACCESS.2021.3062557.

[3] S. Jiang, "Path Planning and Navigation Technologies for Autonomous Mobile Robots in Dynamic Indoor Environments," *Highlights Sci. Eng. Technol.*, vol. 147, pp. 350–358, Jul. 2025, doi: 10.54097/0869dc58.

[4] S. Barlakas *et al.*, "Robot Active Vision-Based Path Planning for Localization Improvement in Indoor Environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2024, pp. 10686–10693. doi: 10.1109/IROS58592.2024.10802041.

[5] T. A. Tola, J. Mi, and Y. qiu Che, "Mapping and Localization of Autonomous Mobile Robots in Simulated Indoor Environments," *Frontiers*, vol. 4, no. 3, pp. 91–100, Sep. 2024, doi: 10.11648/j.frontiers.20240403.13.

[6] J. Zhang, S. Wu, X. Ma, and S. Schwertfeger, "Generation of Indoor Open Street Maps for Robot Navigation from CAD Files," Sep. 08, 2025, *arXiv*: arXiv:2507.00552. doi: 10.48550/arXiv.2507.00552.

[7] R. Alqobali *et al.*, "A Real-Time Semantic Map Production System for Indoor Robot Navigation," *Sensors*, vol. 24, no. 20, Oct. 2024, doi: 10.3390/s24206691.

[8] B. Huang *et al.*, "Development of BIM Semantic Robot Autonomous Inspection and Simulation System," in *2023 9th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, Feb. 2023, pp. 35–40. doi: 10.1109/ICMRE56789.2023.10106602.

[9] A. A. Mahammad, "A Study on the Impact of Building Information Modeling (BIM) in AEC (Architecture, Engineer, Construction) Industry," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 10, pp. 1339–1346, Oct. 2023, doi: 10.22214/ijraset.2023.56202.

[10] X. Zhao and C. C. Cheah, "A Building Information Modeling (BIM)-Enabled Robotic System for Automated Indoor Construction Progress Monitoring," in *2024 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Jun. 2024, pp. 38–43. doi: 10.1109/RCAR61438.2024.10670745.

[11] Ó. M. Mozos, "Semantic Learning of Places from Range Data," in *Semantic Labeling of Places with Mobile Robots*, Ó. M. Mozos, Ed., Berlin, Heidelberg: Springer, 2010, pp. 15–34. doi: 10.1007/978-3-642-11210-2_3.

[12] P. Jensfelt, S. Ekvall, D. Kragic, and D. Aarno, "Augmenting SLAM with Object Detection in a Service Robot Framework," in *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, Sep. 2006, pp. 741–746. doi: 10.1109/ROMAN.2006.314489.

[13] M. Luperto and F. Amigoni, "Predicting the global structure of indoor environments: A constructive machine learning approach," 2019, doi: 10.1007/s10514-018-9732-7.

[14] J. Liu *et al.*, "A BIM Based Hybrid 3D Indoor Map Model for Indoor Positioning and Navigation," *ISPRS Int. J. Geo-Inf.*, vol. 9, no. 12, Dec. 2020, doi: 10.3390/ijgi9120747.

[15] R. Guo, C. Li, and H. Guo, "Indoor Navigation Network Model Construction Method Based on Building Information Model," *J. Geogr. Inf. Syst.*, vol. 15, no. 4, pp. 367–378, Jul. 2023, doi: 10.4236/jgis.2023.154018.

[16] R. G. Braga, S. Karimi, U. Dah-Achinanon, I. Iordanova, and D. St-Onge, "Semantic navigation with domain knowledge," Jun. 18, 2021, *arXiv*: arXiv:2106.10220. doi: 10.48550/arXiv.2106.10220.

[17] J. Zhu *et al.*, "Semantics-based connectivity graph for indoor pathfinding powered by IFC-Graph," *Autom. Constr.*, vol. 171, p. 106019, Mar. 2025, doi: 10.1016/j.autcon.2025.106019.

[18] S. Karimi, R. G. Braga, I. Iordanova, and D. St-Onge, "Semantic Navigation Using Building Information on Construction Sites," Apr. 21, 2021, *arXiv*: arXiv:2104.10296. doi: 10.48550/arXiv.2104.10296.

[19] M. Amani and R. Akhavian, "Safe and trustworthy robot pathfinding with BIM, MHA*, and NLP," *Constr. Robot.*, vol. 9, no. 2, p. 18, Jul. 2025, doi: 10.1007/s41693-025-00161-1.

[20] Z. Chen *et al.*, "Improved coverage path planning for indoor robots based on BIM and robotic configurations," *Autom. Constr.*, vol. 158, p. 105160, Feb. 2024, doi: 10.1016/j.autcon.2023.105160.

[21] M. A. Gopee, S. A. Prieto, and B. García de Soto, "Improving autonomous robotic navigation using IFC files," *Constr. Robot.*, vol. 7, no. 3, pp. 235–251, Dec. 2023, doi: 10.1007/s41693-023-00112-8.

[22] J. Wang, Q. Lu, H. Xu, and H. Zhu, "BIM-to-Robot Mapping: Constructing IFC-Referenced Occupancy Grids and Semantic Metadata for Rule-Based Navigation," in *2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*, Mar. 2025, pp. 780–783. doi: 10.1109/ICAACE65325.2025.11019519.

[23] A. A. Diakite *et al.*, "IFC2INDOORGML: AN OPEN-SOURCE TOOL FOR GENERATING INDOORGML FROM IFC," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIII-B4-2022, pp. 295–301, Jun. 2022, doi: 10.5194/isprs-archives-XLIII-B4-2022-295-2022.

[24] A. A. Diakité, S. Zlatanova, A. F. M. Alattas, and K. J. Li, "TOWARDS INDOORGML 2.0: UPDATES AND CASE STUDY ILLUSTRATIONS," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIII-B4-2020, pp. 337–344, Aug. 2020, doi: 10.5194/isprs-archives-XLIII-B4-2020-337-2020.

[25] E. Chatzinikolaou, I. Pispidikis, and E. Dimopoulou, "A SEMANTICALLY ENRICHED AND WEB-BASED 3D ENERGY MODEL VISUALIZATION AND RETRIEVAL FOR SMART BUILDING IMPLEMENTATION USING CITYGML AND DYNAMIZER ADE," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. VI-4-W1-2020, pp. 53–60, Sep. 2020, doi: 10.5194/isprs-annals-VI-4-W1-2020-53-2020.

[26] T.-A. Teo and S.-C. Yu, "THE EXTRACTION OF INDOOR BUILDING INFORMATION FROM BIM TO OGC INDOORGML," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLII-4-W2, pp. 167–170, Jul. 2017, doi: 10.5194/isprs-archives-XLII-4-W2-167-2017.

[27] L. Sánchez *et al.*, "Data Enrichment Toolchain: A Data Linking and Enrichment Platform for Heterogeneous Data," *IEEE Access*, vol. 11, pp. 103079–103091, 2023, doi: 10.1109/ACCESS.2023.3317705.

[28] S. Gerin, L. Joblot, N. Makhoul, and F. Merienne, "Knowledge Graph as Digital Twins Enhancer for Real Case Data-Driven Smart Building," in *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, T. Borangiu, D. Trentesaux, P. Leitão, and C. Legat, Eds., Cham: Springer Nature Switzerland, 2025, pp. 105–117. doi: 10.1007/978-3-031-85316-6_8.

[29] A. Meucci *et al.*, "Mapping an Information Model for Historic Built Heritage into the IndoorGML Standard: The Case of the Pitti Palace," *Heritage*, vol. 8, no. 4, Mar. 2025, doi: 10.3390/heritage8040115.

[30] P. Mohammed, A. Behan, D. O'Sullivan, and B. McAuley, "Evaluating and Enhancing Georeferencing Accuracy in BIM and 3D GIS Models for Built Environment Digital Twins," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. X-4-W6-2025, pp. 161–168, Sep. 2025, doi: 10.5194/isprs-annals-X-4-W6-2025-161-2025.

[31] Š. Jaud and C. Clemen, "GeoMVD: the Journey to High-Quality Georeferencing Profiles in IFC Datasets," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. X-4-W5-2024, pp. 203–210, Jun. 2024, doi: 10.5194/isprs-annals-X-4-W5-2024-203-2024.

[32] "KIT IFC Examples - IFC Wiki." Accessed: Jan. 05, 2026. [Online]. Available: https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples#IFC_4