FloorPlan2Nav: Semantic-Topological Navigation from Floor Plan Images as Prior Maps

Yan Gao[1,†] Qian Zheng[1,†], Fuji Hu[1], Yiwei Weng [1,*]

[1] Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong
[†]These authors contributed equally to this work and are considered co-first authors
*Corresponding author yiwei.weng@polyu.edu.hk

Abstract –

This paper presents FloorPlan2Nav, a framework that transforms architectural floor plan images into topological navigation graphs serving as prior maps for mobile robots. The framework combines deep learning-based semantic segmentation with contour-based instance extraction, where majority voting within geometrically accurate contours corrects boundary errors in pixel-wise predictions. A connectivity graph constructed from labelled instances enables hierarchical path planning. Graph-guided planning determines the sequence of rooms and connectors, while local A* computes trajectories within each segment in parallel. Experiments on the ResPlan dataset demonstrate 98.16% instance-level segmentation accuracy and 56.89% reduction in planning time compared to global A*. Simulation also confirms that floor plans provide sufficient geometric constraints to correct positional drift, enabling successful navigation without SLAM-based exploration.

Keywords –

Floor Plan Parsing; Connectivity Graph; Hierarchical Path Planning; Mobile Robot Navigation; Prior Map

1 Introduction

Autonomous indoor navigation requires environment maps for path planning and localisation. Conventional approaches rely on SLAM to construct such maps, but the robot must first explore the environment before it can begin operation. Recent studies have explored using prior maps derived from building documentation to reduce this exploration effort. BIM-guided frameworks construct navigation maps from IFC data [1], [2], and visual SLAM systems have incorporated BIM constraints to reduce trajectory drift [3]. These approaches are effective when BIM models are available, but many buildings do not have such documentation. Floor plan images are far more accessible. They can be found in real estate listings, facility archives, and evacuation signs, and they contain the geometric information about walls, rooms, and doors needed for navigation. However, unlike BIM or IFC data, floor plan images are unstructured, and the spatial information is embedded in pixels rather than represented as geometric entities with semantic labels. Hence, a method that can automatically parse floor plan images into structured navigation maps would enable prior map-based indoor navigation in a wider range of buildings.

This paper presents FloorPlan2Nav, a framework that transforms floor plan images into topological navigation graphs. The framework combines semantic segmentation with contour-based instance extraction to identify functional spaces and connectors such as doors, windows, and front doors with precise boundaries. These instances are organised into a connectivity graph that can be adapted to different robot types by enabling edges according to their traversal capabilities. Path planning operates hierarchically, i.e., graph-guided planning first determines the sequence of rooms and connectors, then local A* computes trajectories within each segment in parallel, which significantly reduces planning time. The framework is evaluated on the ResPlan dataset [4] and validated in simulation. Results demonstrate improved planning efficiency and show that floor plans can also serve as geometric reference to correct positional drift during navigation.

2 Related Work

Floor plan understanding has been extensively studied using deep learning methods. Datasets such as CubiCasa5K [5] and ResPlan [4] provide thousands of annotated floor plans covering various room types and architectural styles. ResPlan is particularly relevant to this work as it includes explicit room adjacency graphs alongside vector representations, enabling both geometric and topological analysis. Using these datasets, segmentation networks can effectively learn to classify walls, rooms, doors, and windows at the pixel level [5], [6]. However, navigation relies on topological graphs where spatial adjacency is typically detected through morphological operations on segmented masks. Pixel-wise accuracy is unable to guarantee correct topology, because predictions often overflow or fall short of true boundaries. Such errors lead to incorrect adjacency relationships, which is problematic for downstream path planning.

Prior map-based navigation has been explored using structured building data. Chen et al. [7] developed a BIM-integrated path planning framework using IFC-based layered mapping and adaptive A* for ground robots. Blum et al. [8] proposed LiDAR localisation in 3D building models using semantic filtering and ICP registration. Gao and Kneip [9] achieved drift-free localisation using CAD-based floor plans as reference maps. More recently, graph-based representations have gained attention for their efficiency and semantic expressiveness. Zhu et al. [10] demonstrated that connectivity graphs constructed from IFC data enable efficient pathfinding by exploiting spatial relationships rather than dense geometry. Ayanzadeh and Oates [11] explored using large language models (LLMs) to parse floor plan images into spatial knowledge graphs for navigation assistance.

While these methods advance prior map-based navigation, they rely on structured building data such as BIM, IFC, or CAD, or require robots to pre-explore environments to construct spatial representations. However, in practice, such digital documentation or pre-exploration is often unavailable. By contrast, floor plan images are readily accessible from real estate listings, evacuation signs, and facility archives. This paper aims to address the challenge of automatically generating navigation-ready topological graphs from floor plan images, thereby bridging the gap between readily available architectural imagery and robot navigation systems.

3 Methodology

The proposed FloorPlan2Nav framework transforms architectural floor plan images into topological navigation graphs through three main stages: semantic segmentation with instance extraction, connectivity graph construction, and hierarchical path planning.

3.1 Semantic Segmentation and Instance Extraction

The input to our framework is a blank architectural floor plan without any semantic annotation. Semantic segmentation is employed to transform the raw drawing into a coloured semantic map. A DeepLabV3+ network trained on the ResPlan dataset [4] learns the visual and geometric patterns of residential floor plans and predicts pixel-wise labels for walls, rooms, doors, windows, and other architectural elements. However, pixel-wise segmentation often suffers from local boundary errors where predicted masks overflow beyond true room boundaries or fall short of them, leaving small gaps. To obtain geometry-accurate instances, we perform contour-based extraction directly on the original floor plan drawing. The image is binarised to isolate structural drafting lines that depict walls and boundaries. These lines naturally partition the plane into enclosed regions. Connected-component labelling identifies all enclosed regions that do not touch the image boundary, and a contour-tracing algorithm extracts the precise boundary of each region (see Figure 1).

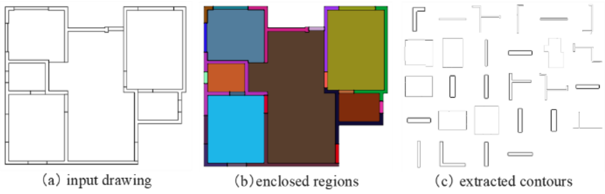


Figure 1 Instance extraction from floor plan

With all instances extracted, the semantic label of each instance is determined through majority voting over the predicted pixel labels within its contour. This effectively corrects local boundary errors in the raw semantic predictions. As shown in Figure 2, when the semantic mask falls short of the true boundary or overflows into adjacent regions (indicated by dashed boxes), the majority vote still assigns the correct label based on dominant predictions inside the geometrically accurate contour. This hybrid approach combines the semantic recognition capability of deep learning with precise geometric boundaries guaranteed by contour-based extraction, which is required for correct graph construction.

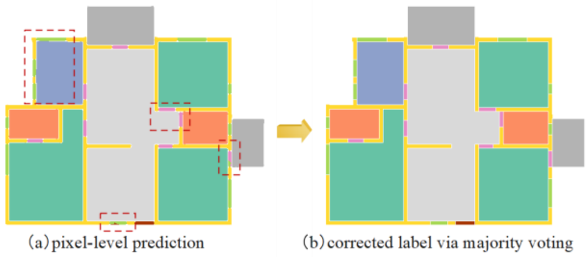


Figure 2 Semantic label assignment

3.2 Connectivity Graph Construction

The annotated floor plan is converted into a connectivity graph for navigation. Nodes correspond to segmented instances grouped into functional spaces (living rooms, bedrooms, bathrooms, kitchens) and connectors (doors, windows, front doors, and open balconies). Walls are excluded as they represent barriers rather than navigable connections. Spatial adjacency is detected by applying morphological dilation with kernel size k to each instance mask. Overlapping dilated boundaries indicate potential connections, and since each instance is processed independently, this operation can run in parallel. However, not all overlaps represent valid connectivity. Two adjacent doors may overlap after dilation but do not form a traversable path. Semantic filtering removes such connector-to-connector links and retains only space-to-connector relationships (see Figure 3).

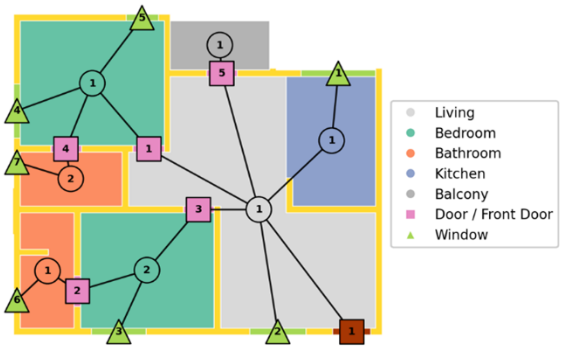


Figure 3 Connectivity graph construction

Which connectors are traversable depends on the robot's physical capabilities. Ground robots can only pass through doors and front doors. UAVs can additionally traverse windows and open balconies, enabling exterior routes along the building facade. Edge weights can further encode traversal costs such as penalties for stairs or narrow passages, making the graph adaptable to different robot types.

3.3 Hierarchical Path Planning

The connectivity graph enables a two-stage planning approach. Global path planning searches the graph to determine the sequence of rooms and connectors the robot should traverse. Figure 4 illustrates alternative routes from balcony to front door for different robot types. Ground robots (a) can only pass through doors, while UAVs (b) can also traverse windows and fly along the building exterior.

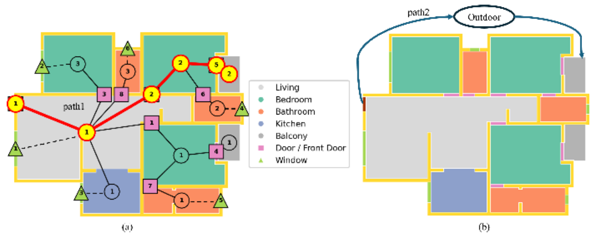


Figure 4 Path alternatives: (a) ground robot, (b) UAV

With the path determined, local path planning computes the trajectory within each segment using A*. Connector centroids are used as waypoints to ensure safe passage through doorways. Since segments between consecutive connectors are independent, local planning runs in parallel (see Figure 5), and computational savings increase with the number of segments, making the approach scalable to larger and more complex floor plans.

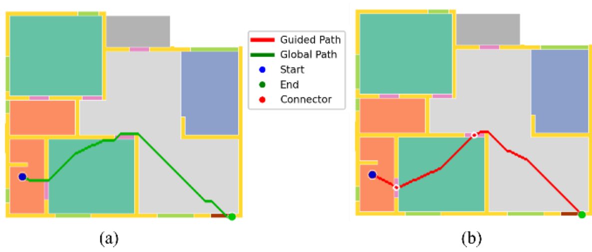


Figure 5 Path planning: (a) global A*, (b) segment-based parallel approach

4 Experiment

4.1 Experiment Setup

We evaluate our framework on the ResPlan dataset, which contains 17,107 residential floor plans with pixel-level annotations for 10 semantic classes: background, living room, bedroom, bathroom, kitchen, balcony, wall, door, window, and front door. The dataset is split into training, validation, and test sets with a ratio of 8:1:1. For semantic segmentation, we employ DeepLabV3+ [12] with Xception-65 backbone pre-trained on PASCAL VOC 2012. The network is trained for 100 epochs using SGD with momentum 0.9, learning rate decaying from 7×10^{-3} to 7×10^{-5} with cosine annealing, and input size of 512×512 pixels. The dilation kernel size k for graph construction is set to 5 pixels. Path planning experiments are conducted on Intel Core i7-11800H (2.30GHz) with 16GB RAM.

4.2 Segmentation Results

We evaluate the model's ability to predict semantic labels from blank floor plan drawings. The DeepLabV3+ model achieves 95.86% mean IoU and 99.54% pixel accuracy. After majority voting within instance contours, instance-level classification reaches 98.16% accuracy across 55,065 instances from 1,711 test images. Table 1 shows results grouped by category. Functional spaces and walls achieve over 99% F1 score. Connectors show lower accuracy due to visual confusion among door, window, and front door. However, for navigation the key requirement is distinguishing connectors from non-traversable elements rather than differentiating connector types. Table 2 shows that when treated as a binary classification, connector detection reaches 99.75% precision and 99.60% recall.

Table 1 Instance-level segmentation results

Category	Precision	Recall	F1
Functional spaces	99.21%	99.03%	99.12%
Connectors	96.48%	96.03%	96.25%
Wall	99.34%	99.56%	99.45%
Background	96.11%	97.93%	97.01%
Overall	98.00%	98.16%	98.08%

Table 2 Connector confusion matrix

GT \ Pred	Connector	Others
Connector	23,882	97
Others	61	31,025

4.3 Path Planning Efficiency

We compare the proposed graph-guided approach against global A* on path planning efficiency. 100 floor plans are randomly selected from the test set, and 10 start-goal pairs are generated for each floor plan by randomly sampling functional spaces, yielding 1,000 navigation tasks in total. Table 3 reports the average performance across all tasks.

The graph-guided method reduces planning time by 56.89% and explores 58.13% fewer nodes compared to global A*. The path cost increases by 7.34% because routing through door centres produces slightly longer trajectories than paths that cut close to door edges. In practice, this difference is negligible since robots adjust trajectories dynamically during execution.

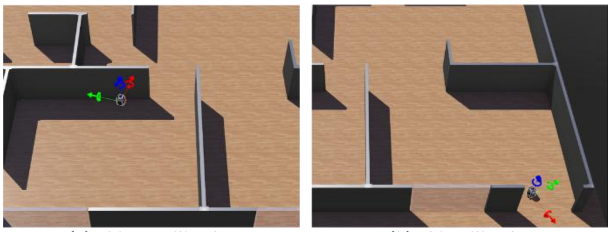
While the absolute time savings are small for these residential layouts, the hierarchical design is valuable because of how it scales. Global A* must search across the entire map grid, so planning time increases as the floor plan grows larger. In contrast, graph-guided planning only searches within individual room segments, each of which stays small regardless of overall building size. Since these segments are computed in parallel, planning time does not accumulate as the number of rooms increases. Therefore, the advantage is greatest in larger commercial or institutional floor plans.

4.4 Simulation Validation

We conduct simulation experiments in Webots to test navigation using the floor plan as a prior map. Webots is an open-source robot simulator that provides realistic physics and sensor models. We use a simulated TurtleBot3 Burger equipped with LDS-01 LiDAR (360° FOV, 0.12-3.5m range). The setup assumes no discrepancy between the floor plan and physical environment, and rooms are empty. The robot follows waypoints generated from the connectivity graph. To simulate realistic localisation drift, Gaussian noise (σ=20mm per timestep) is injected into XY odometry while yaw is assumed accurate from IMU. Without using the floor plan for position correction, cumulative drift causes the robot to deviate and collide with walls (Figure 6a). With periodic calibration, the robot completes the trajectory successfully (Figure 6b).

Table 3 Path planning performance comparison

Metric	Global A*	Graph-guided (ours)	Improvement
Planning time (ms)	23.42	10.09	-56.89%
Nodes explored	2,231	934	-58.13%
Path cost (px)	85.29	91.57	7.34%



(a) without calibration (b) with calibration

Figure 6 Navigation results in Webots simulation

Calibration is triggered every five waypoints. The robot extracts line segments from LiDAR scans and matches them against visible wall contours from the floor plan. Matching candidates are filtered by segment orientation (horizontal or vertical) and validated by distance and angular proximity relative to the robot. Since residential layouts are typically axis-aligned, corrections are decoupled along orthogonal directions: horizontal segments refine Y position and vertical segments refine X.

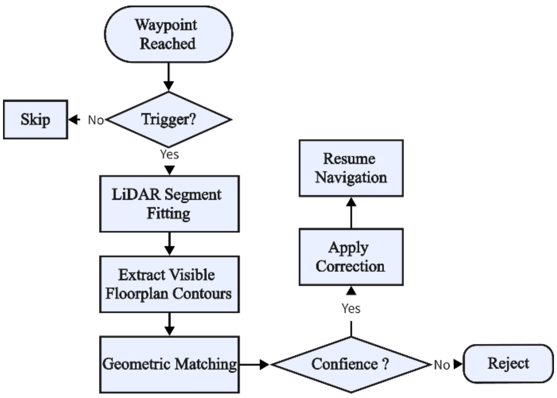


Figure 7 LiDAR calibration flowchart

The matching process and calibration results are illustrated in Figures 8 and 9. Figure 8 shows the segment matching at four waypoints, where segments extracted from LiDAR scans (solid lines) are matched against wall contours from the floor plan (dashed lines). The number of matched pairs varies depending on visible walls. Figure 9 shows the corresponding calibration results. After position correction, the robot position (blue) moves consistently closer to ground truth (green) compared to the drifted estimate (red), confirming that floor plan geometry provides adequate reference for odometry compensation.

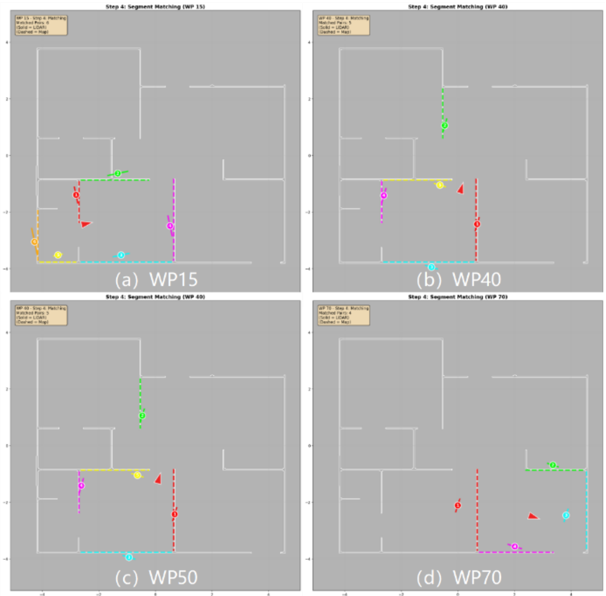


Figure 8 Segment matching at four waypoints

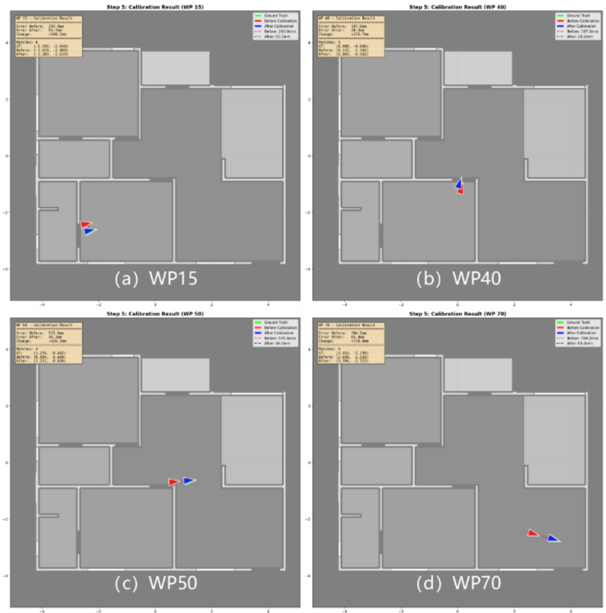


Figure 9 Calibration results at four waypoints

5 Discussion

This work has several limitations. Firstly, the simulation assumes no discrepancy between the floor plan and physical environment, which may not hold for older buildings with undocumented renovations. Secondly, the calibration scheme relies on axis-aligned wall geometry and accurate yaw from IMU. Rooms are also assumed to be empty without furniture that could occlude walls or block paths. The framework also operates on 2D floor plan images only and does not incorporate three-dimensional information such as floor slopes, door sill heights, or elevation changes. Addressing these factors would require additional elevation data or depth sensing, which is beyond the scope of this work.

The pipeline relies on accurate segmentation, and the sensitivity of navigation to segmentation errors has not been analysed. The contour-based instance extraction and majority voting correct minor pixel-level errors, but more severe failures such as a missed door or a merged room boundary could disrupt the connectivity graph. Furthermore, floor plan segmentation is a well-studied problem, and high accuracy has been demonstrated on both furnished and unfurnished floor plans [13]. The 98.16% instance accuracy on the ResPlan test set is consistent with this. A systematic robustness analysis is left for future work.

Future work will address building discrepancies through probabilistic matching methods that handle geometric deviations. Another direction is vision-language navigation (VLN), where the connectivity graph serves as spatial knowledge rather than a source of precomputed waypoints. Instead of relying on precise geometric matching, the robot observes its surroundings, matches what it sees against the expected layout from the floor plan, and moves toward the next connector by visually identifying doors and walls in real-time, allowing the robot to handle environmental changes and partial occlusions. In this setting, the floor plan serves not as a geometric reference for contour matching, but as a semantic map providing spatial knowledge such as room types and connectivity.

6 Conclusion

This paper presents FloorPlan2Nav, a framework that transforms blank floor plan images into navigation-ready representations for mobile robots. The floor plan is utilised at three stages. Semantic segmentation combined with contour-based instance extraction converts the raw drawing into labelled functional spaces and connectors, where majority voting within geometrically accurate contours corrects boundary errors in pixel-wise predictions. A connectivity graph constructed from these instances enables hierarchical path planning, i.e., global planning on the graph determines the sequence of rooms and connectors, while local A* computes trajectories within each segment. Since each segment is independent, they can be computed in parallel, and the efficiency gain scales with building complexity. The floor plan also serves as a prior map for localisation during navigation, which allows robots to correct odometry drift by matching LiDAR observations against known wall geometry.

Experiments on the ResPlan dataset demonstrate 98.16% instance-level segmentation accuracy with 99.60% connector detection recall. The graph-guided planning achieves 56.89% reduction in computation time compared to global A*. Simulation in Webots confirms that floor plan-based calibration effectively corrects accumulated drift and enables successful navigation without SLAM-based exploration.

Acknowledgement

The authors would like to gratefully acknowledge the support from the Hong Kong Polytechnic University (P0051072 and P0058004).

References

[1] Z. Chen, X. Liang, J. Yuan, S. Yang, and K. Chen. Automated reality capture for indoor inspection using BIM and a multi-sensor quadruped robot. *Automation in Construction*, 170, 2025. doi: 10.1016/j.autcon.2024.105930.

[2] M. A. V. Torres, A. Braun, and A. Borrmann. BIM-SLAM: Integrating BIM models in multi-session SLAM for lifelong mapping using 3D LiDAR. In *Proceedings of the 41st International Symposium on Automation and Robotics in Construction (ISARC)*, pages 521–528, Lille, France, 2023. doi: 10.22260/ISARC2023/0070.

[3] A. Bikandi-Noya, M. Fernandez-Cortizas, M. Shaheer, A. Tourani, H. Voos, and J. L. Sanchez-Lopez. BIM informed visual SLAM for construction monitoring. On-line: http://arxiv.org/abs/2509.13972, Accessed: 30/01/2026.

[4] M. Abouagour and E. Garyfallidis. ResPlan: A large-scale vector-graph dataset of 17,000 residential floor plans. On-line: http://arxiv.org/abs/2508.14006, Accessed: 30/01/2026.

[5] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala. CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis. On-line: http://arxiv.org/abs/1904.01920, Accessed: 30/01/2026.

[6] Z. Fan, L. Zhu, H. Li, X. Chen, S. Zhu, and P. Tan. FloorPlanCAD: A large-scale CAD drawing dataset for panoptic symbol spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10128–10137, 2021.

[7] Z. Chen, K. Chen, and J. C. P. Cheng. BIM-integrated indoor path planning framework for unmanned ground robot. In *Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 776–783,

Dubai, UAE, 2021. doi: 10.22260/ISARC2021/0105.

[8] H. Blum, J. Stiefel, C. Cadena, R. Siegwart, and A. Gawel. Precise robot localisation in architectural 3D plans. On-line: http://arxiv.org/abs/2006.05137, Accessed: 30/01/2026.

[9] L. Gao and L. Kneip. FP-Loc: Lightweight and drift-free floor plan-assisted LiDAR localisation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4142–4148, 2022. doi: 10.1109/ICRA46639.2022.9812361.

[10] J. Zhu, X. Wang, M. Wang, P. Wu, H. Kim, and J. Wang. Semantics-based connectivity graph for indoor pathfinding powered by IFC-Graph. *Automation in Construction*, 171, 2025. doi: 10.1016/j.autcon.2025.106019.

[11] A. Ayanzadeh and T. Oates. Floorplan2Guide: LLM-guided floorplan parsing for BLV indoor navigation. On-line: http://arxiv.org/abs/2512.12177, Accessed: 30/01/2026.

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[13] B. Yang, H. Jiang, Z. Lu, and J. Xiao. Exploring structural lines for interior floorplan segmentation. *The Visual Computer*, 41:6983–6997, 2025. doi: 10.1007/s00371-025-04037-x.