

Article

Development of Navigation Network Models for Indoor Path Planning Using 3D Semantic Point Clouds

Jiwei Hou , Patrick Hübner and Dorota Iwaszczuk * 

Remote Sensing and Image Analysis, Department of Civil and Environmental Engineering, Technical University of Darmstadt, 64287 Darmstadt, Germany; jiwei.hou@tu-darmstadt.de (J.H.)

* Correspondence: dorota.iwaszczuk@tu-darmstadt.de

Abstract: Accurate and efficient path planning in indoor environments relies on high-quality navigation networks that faithfully represent the spatial and semantic structure of the environment. Three-dimensional semantic point clouds provide valuable spatial and semantic information for navigation tasks. However, extracting detailed navigation networks from 3D semantic point clouds remains a challenge, especially in complex indoor spaces like staircases and multi-floor environments. This study presents a comprehensive framework for developing and extracting robust navigation network models, specifically designed for indoor path planning applications. The main contributions include (1) a preprocessing pipeline that ensures high accuracy and consistency of the input semantic point cloud data; (2) a moving window algorithm for refined node extraction in staircases to enable seamless navigation across vertical spaces; and (3) a lightweight, JSON-based storage structure for efficient network representation and integration. Additionally, we presented a more comprehensive sub-node extraction method for hallways to enhance network continuity. We validated the method using two datasets—the public S3DIS dataset and the self-collected HoloLens 2 dataset—and demonstrated its effectiveness through Dijkstra-based path planning. The generated navigation networks supported practical scenarios such as wheelchair-accessible path planning and seamless multi-floor navigation. These findings highlight the practical value of our approach for modern indoor navigation systems, with potential applications in smart building management, robotics, and emergency response.

Keywords: indoor navigation; semantic point clouds; navigation network model; path planning



Academic Editors: Shengcheng Yeh,
Yih-Shyh Chiou and Bang Wang

Received: 17 December 2024

Revised: 16 January 2025

Accepted: 20 January 2025

Published: 23 January 2025

Citation: Hou, J.; Hübner, P.; Iwaszczuk, D. Development of Navigation Network Models for Indoor Path Planning Using 3D Semantic Point Clouds. *Appl. Sci.* **2025**, *15*, 1151. <https://doi.org/10.3390/app15031151>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rise in advanced technologies like Internet of Things (IoT) and augmented reality (AR), there is a growing demand for efficient and accurate indoor navigation systems that can support a wide range of applications [1–3], from guiding users through complex buildings to ensuring safe, wheelchair-accessible pathways. At the core of these systems is the need for precise indoor navigation models, which represent the spatial and semantic structure of indoor environments.

In recent years, various indoor navigation models have been developed, mainly including three types [4,5]: the network model [6–9], the grid/voxel model [10,11], and the space entity model [12]. These models, along with their development methods, each have unique advantages and limitations in representing complex indoor environments, making it essential to choose both the appropriate model and construction method based on the specific requirements of different navigation applications, such as low cost, high

efficiency in development, and robust performance in real-world applications. For the network model, its simplicity, flexibility, and efficiency have made it widely favored for constructing indoor navigation systems. Moreover, recent advancements in 3D mapping technologies, along with the development of deep learning, have opened new avenues for the automatic extraction of indoor navigation network models [2,13,14].

In the field of 3D mapping, the availability of low-cost sensor devices such as the Apple smart devices [15,16], Microsoft HoloLens [17–19], and depth cameras [20–22] allows for an increasing number of users, even those without professional knowledge, to perform 3D mapping and obtain indoor spatial information [15]. This not only reduces the cost of indoor mapping, but also allows more non-experts to participate. Furthermore, with the rapid development of deep learning technologies [23–25], it has become possible to train 3D point cloud semantic segmentation models that can automatically recognize and classify various elements as semantic point clouds within indoor spaces with a reasonable degree of accuracy. Semantic point clouds, which contain detailed geometric and semantic information about indoor environments, offer significant potential for extracting indoor navigation elements. However, their quality can vary substantially due to differences in data acquisition methods, annotation techniques, and segmentation models, often resulting in inconsistencies, noise, and incomplete semantic labels. Consequently, extracting accurate navigation networks from such point clouds often requires manual intervention during the preprocessing stage. However, to the best of our knowledge, no standardized preprocessing framework currently exists to guide this process.

While research on extracting indoor navigation networks from semantic point clouds remains limited, some progress has been made. For example, Yang et al. [26] proposed a semantics-guided method for reconstructing indoor navigation networks directly from RGB-D sensor data. However, their research does not fully address the challenges posed by multi-floor environments, particularly the accurate representation of transitional spaces such as staircases and elevators. The transitional spaces are critical for seamless vertical connectivity and play a crucial role in various real-world applications. For example, in robotics, an accurate navigation network of such spaces facilitates autonomous navigation across multiple floors. Additionally, in emergency response systems, precise information about transitional spaces enables responders to quickly navigate complex indoor environments, thereby improving the success rate of rescue operations.

Specifically, staircases present a notable challenge in accurately extracting the direction and layout of staircases in an automated manner due to their complex geometries, variable step structures, and possible direction changes, especially for unstructured point cloud data. While the current method can automatically identify staircase areas to some extent, representing staircase areas with a single node or a single straight line in the navigation network fails to accurately reflect the actual paths of staircases [6,27,28]. And manually extracted detailed staircase nodes [5] are time-consuming and may not be suitable for large-scale indoor navigation network extraction. Using Building Information Modeling (BIM) data, Qiu et al. [29] proposed an automatic method to generate comprehensive 3D indoor maps, integrating both floor-level and cross-floor paths through a hybrid grid-topological map approach. Their method effectively adapts to complex building geometries and achieves high accuracy rates in extracting paths across multi-floor structures. However, compared to structured BIM data, point cloud data can be more directly acquired, developing an automatic and efficient method to accurately capture the actual paths of staircases from 3D point cloud and adapt to different staircase designs still requires further research.

Therefore, the main challenges in accurately extracting indoor navigation networks from semantic point clouds can be summarized into two key aspects: (1) developing a standardized preprocessing framework to ensure the quality and reliability of the data

for navigation network extraction; and (2) effectively representing complex multi-floor transitional spaces, particularly staircases, in a way that accurately and efficiently captures their true paths and layouts.

To address the challenges above, in this work, we mainly focused on the following aspects:

- (1) Introducing a preprocessing rule to standardize and refine the semantic point cloud data, ensuring its accuracy and consistency for reliable navigation network extraction.
- (2) Developing a moving window method to ensure that the extracted navigation network aligns more closely with the actual structure of staircases.
- (3) Designing a lightweight data storage structure in the JavaScript Object Notation (JSON) format [30] specifically to store navigation network information extracted from 3D semantic point clouds.

Additionally, for indoor navigable space classification, this work referenced the IndoorGML standard [31], which divides navigable space into general space, connection space, and transition space. For experimental evaluation, in this study, besides using the public Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [32], we further validated the effectiveness and robustness of our indoor navigation network extraction algorithms on our own dataset.

In the following, Section 2 outlines the methodology for extracting JSON-encoded navigation networks from 3D semantic point clouds, including data preprocessing, space classification, and navigation network generation. Section 3 presents the experimental process and results. Section 4 provides a discussion and analysis of the results, and Section 5 concludes the study and suggests future work.

2. Methodology

The goal of this study is to develop an automatic algorithm to extract indoor navigation networks from 3D semantic point clouds. Figure 1 illustrates the pipeline of the proposed method. In the following sections, each step of the proposed method will be introduced in detail.

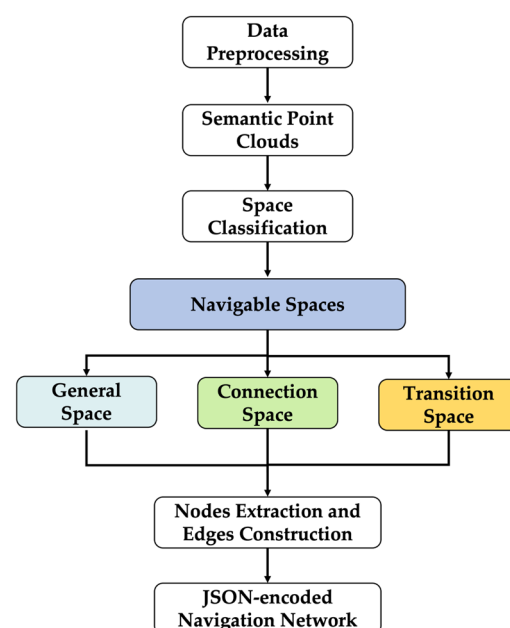


Figure 1. The pipeline for extracting JSON-encoded navigation network for indoor path planning from 3D semantic point clouds.

2.1. Data Preprocessing

In this study, the purpose of data preprocessing is to establish specific rules for generating standardized semantic point clouds that are suitable for our indoor navigation network extraction algorithm, while also ensuring consistency in the file storage structure.

In this study, to ensure that the semantic point cloud data input to the algorithm has good quality, we conducted data preprocessing floor-wise and manually segmented the data to obtain the required semantic point clouds. All data segmentation and semantic labeling are conducted in CloudCompare software, version 2.13 [33]. In association with the space classification in Section 2.2, we conducted data preprocessing according to the characteristics of different navigation spaces, with a particular focus on processing hallways and stairwells, as well as the supplementation of virtual doors. Additionally, the preprocessing step effectively removed noise points from the data, ensuring higher accuracy and reliability of the semantic point clouds.

Hallways are important passages connecting different rooms and other indoor areas. Thus, the accurate segmentation and labeling of hallways are important for indoor navigation element extraction and network construction. Our preprocessing of hallways mainly focuses on those with turns (i.e., close to 90°) in the original data. If a hallway has turns along its centerline, it is treated as a composite hallway. In such cases, composite hallways could complicate the indoor navigation element extraction and construction of navigation networks. Therefore, we segmented these composite hallways into multiple simple, straight hallways without turns, as illustrated in Figure 2.

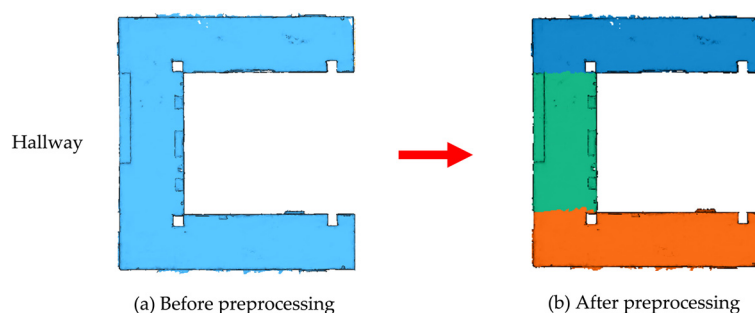


Figure 2. The illustration of a composite hallway converted into multiple simple, straight hallways without turns.

In general, stairwells consist mainly of the staircase and landing. The staircase is an important component that connects the topological relationship between different floors. In this study, we mainly preprocessed the data for the staircase part and extracted the staircase elements from the stairwell individually. Additionally, we also extracted the elevator element from the elevator lobby. The illustration of the extracted staircase and elevator is shown in Figure 3.

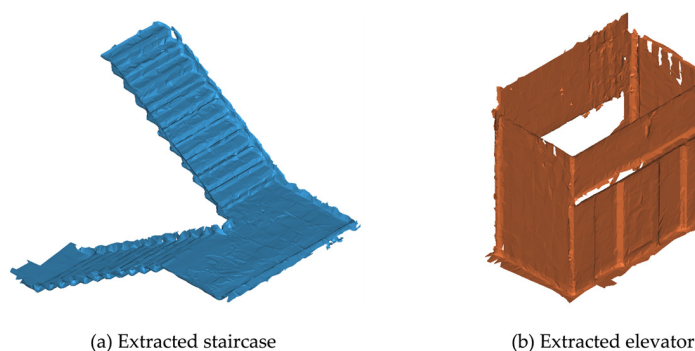


Figure 3. The illustration of the extracted staircase and elevator.

Furthermore, we also extracted the doors corresponding to each type of space, such as the doors leading to office rooms or those associated with hallways. It is important to note that in indoor spaces, there are not only physical doors but also walkable shared boundaries. For example, the walkable shared boundary between two adjacent hallways or the walkable shared boundary between a room and a hallway without a physical door. In the research of Yang et al. [26], they defined these walkable shared boundaries as virtual doors. Although these virtual doors do not look like physical doors, they play an equally important role in connecting adjacent indoor spaces.

In our study, we adopted this definition, using virtual doors to connect adjacent indoor spaces that have walkable shared boundaries. Users can also move from one space to another through these virtual doors. These virtual doors, along with physical doors, are part of the connection space. Through data preprocessing, we individually extracted these doors and virtual doors and saved them in their respective spatial directories. As an example, the extracted doors and virtual doors from the hallway are illustrated in Figure 4. It is worth noting that, because our virtual doors are manually segmented, we did not extract the virtual doors using a uniform thickness standard during the segmentation process. As a result, in some cases, the thickness of the virtual doors may vary. However, this difference does not affect the primary function of the virtual doors in the navigation network, which is to connect adjacent spaces that lack physical doors but have walkable shared boundaries.

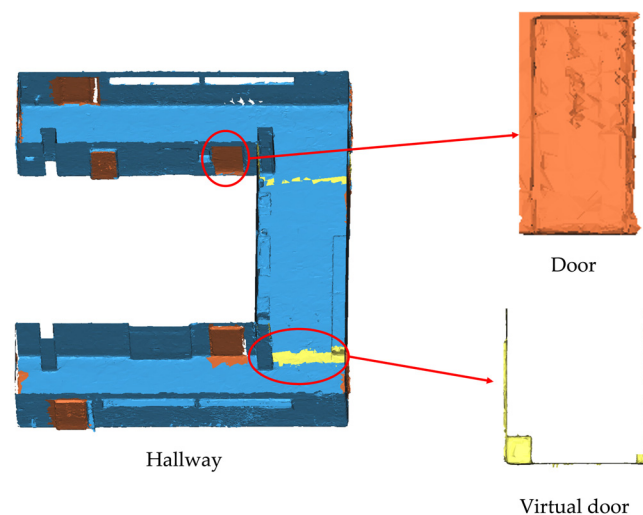


Figure 4. The illustration of the extracted doors and virtual doors from the hallway.

Additionally, to ensure the consistency of the file storage structure, all data are saved following a uniform file tree structure. For example, in the hypothetical Floor 1 scene of a building, this scene includes 2 hallways, 1 stairwell, 1 elevator lobby, 1 office, and 1 conference room. The corresponding file tree for the semantic point clouds in this scene is shown in Figure 5. Taking stairwell_1 as an example, stairwell_1.txt corresponds to the point cloud file for the stairwell, while the annotations folder contains the point cloud files for the doors and staircase associated with the stairwell. Through data preprocessing, we obtained the refined semantic point cloud files, ensuring that the input files to the algorithm follow the same directory structure.

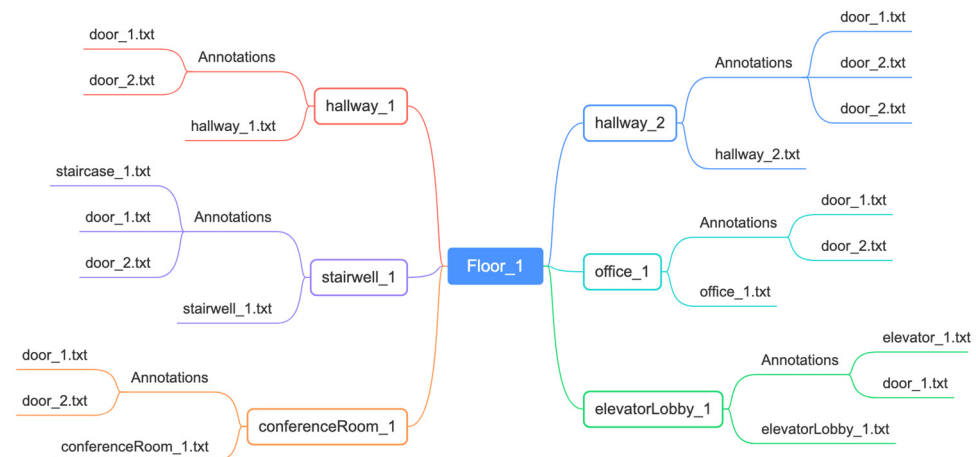


Figure 5. An example of the file tree structure after data preprocessing.

2.2. Space Classification

For the classification of point clouds, we adopted the key concept of navigable space classification from the IndoorGML standard [31]. We classified the point clouds into general space (e.g., offices, conference rooms, and storage), connection space (e.g., doors and virtual doors) and transition space (e.g., hallways and stairwells), as shown in Table 1. These three space categories together form the navigable space, which is an important component of the indoor navigation network.

Table 1. The semantic point clouds space classification.

Space	Category of Space	Components
Navigable space	General space	e.g., office, conference room, storage, copy room, open space, pantry, lounge, lobby, and WC
	Connection space	door and virtual door
	Transition space	hallway, stairwell, and elevator lobby

However, the number of space categories can be adjusted based on specific applications or objectives. For example, if it is necessary to highlight certain areas such as building entrance, emergency exit, or water closet (WC), additional space types can be introduced. In this work, the detailed connectivity provided by connection spaces and transition spaces is sufficient for constructing an effective indoor navigation network, so we classified the point clouds into just these three categories.

2.3. Node Extraction and Edge Construction

In this section, we detail the process of extracting nodes from 3D semantic point clouds. Nodes representing spaces with connected topological relationships are subsequently linked to form edges. The nodes and edges form the backbone of our indoor navigation network, representing key locations and the paths connecting them.

The theoretical foundation for our approach relies on two key concepts: The Node-Relation Graph (NRG) [6] and Poincaré duality [34]. The NRG is a framework representing topological relationships, such as adjacency and connectivity, among indoor objects. It implements these relationships as a graph without requiring geometrical properties, enabling efficient path planning in indoor navigation systems. Poincaré duality provides the theoretical basis for mapping indoor spaces to an NRG. According to Poincaré duality, a k -dimensional object in N -dimensional primal space maps to an $(N-k)$ dimensional object in dual space. For example, 3D rooms are represented as 0D nodes, and 2D surfaces between

rooms become 1D edges. This simplification allows for a combined topological network model that effectively manages complex spatial relationships indoors.

Due to the different structural features, the importance of different types of navigable spaces is also different, e.g., hallways connecting many rooms and staircases connecting different floors are more important than general and connection spaces. Therefore, the node extraction strategies for general space, connection space, and transition space are slightly different.

For each semantic point cloud representing general, connection, or some transition spaces (e.g., elevator lobbies), we extract the centroid to serve as the node corresponding to each space in the NRG. For a group of points $P = \{p_1, p_2, \dots, p_n\}$ in a semantic point cloud (e.g., offices, doors), where $p_i = (x_i, y_i, z_i)$, the centroid is calculated as

$$\text{Centroid} = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i, \frac{1}{n} \sum_{i=1}^n z_i \right) \quad (1)$$

However, for staircases and hallways, which belong to transition space, a single representative node sometimes fails to accurately reflect the geometric relationships, making it impossible to generate the shortest paths. Since hallways could be long and connect many rooms, and staircases link different floors, it is necessary to refine the nodes for both hallways and staircases. In the following, we will detail the extraction of nodes for staircases and hallways in Section 2.3.1 and Section 2.3.2, respectively.

2.3.1. Node Extraction and Edge Construction for Staircases

In this study, considering the vertical height changes and the more complex 3D structure of staircases, we designed a dedicated algorithm for extracting the nodes and edges of staircases for each floor.

The first step involves extracting the centerline of the staircase, which is achieved using a moving window approach along the Z-axis, as illustrated in Figure 6. This method calculates the bounding box of the point cloud data representing the staircase to understand its spatial extents. The point cloud data are then processed in slices incrementally along the Z-axis. For each slice within the bounding box, we identify the points that fall within the current Z-range and calculate their centroid using Equation (1). These centroids form the centerline points of the staircase.

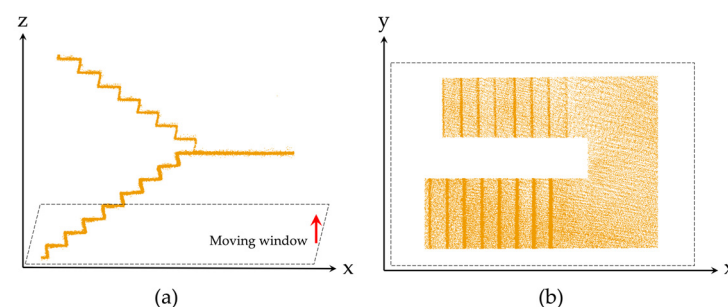


Figure 6. An illustration of the moving window approach for staircase nodes extraction. The red arrow indicates the moving direction along the Z-axis: (a) side view, (b) top view.

To ensure the extracted centerline is smooth and continuous, we adjust each point on the centerline by considering the positions of its neighboring points, resulting in a more regular representation of the staircase path.

Specifically, given a sequence of centerline points $\{P_i\}$, where $i = 0, 1, \dots, N$ represents the index of each point along the centerline, for each interior point P_i (i.e., excluding

the first and last points), the new smoothed position P'_i is calculated by blending the current point with the average of its neighboring points:

$$P'_i = (1 - \alpha)P_i + \alpha \frac{P_{i-1} + P_{i+1}}{2} \quad (2)$$

In Equation (2), P_i is the current point being smoothed and P_{i-1} and P_{i+1} are the previous and next points, respectively. α is the smoothing factor, controlling the influence of neighboring points. A smaller value of α results in less smoothing, while a larger value increases the smoothing effect.

Once the centerline is extracted and smoothed, nodes are created at each point along the smoothed centerline, each assigned an ID and labeled as a staircase node, with coordinates reflecting its position on the centerline. Edges are created between consecutive nodes along the centerline, establishing a connected path. Additionally, nodes representing doors in the stairwell space are connected to the first node of the staircase.

To maintain consistency in the vertical dimension, the Z-values of non-staircase nodes are adjusted to match the minimum Z-value of the staircase nodes. This ensures that all extracted non-staircase nodes on the same floor have a uniform height with respect to the minimum Z-value of the staircase node. We first extract the navigation network structure for each floor separately and then connect these structures through the staircase nodes.

2.3.2. Node Extraction and Edge Construction for Hallways

For a more practical extraction of a navigation route in hallways, Jianga et al. [35] proposed a method for subdividing corridors based on the positions of doors. By calculating the cutting points between doors and projecting perpendicular lines, they divide the hallway into several segments, thereby generating a more detailed and accurate navigation route. However, this method relies on 2D CAD files for navigation network generation and may not be directly applicable to 3D point clouds. In the study by Yang et al. [26], they used a method for extracting sub-nodes along the hallway. In their method, one key step is determining the hallway centerline, followed by projecting the positions of doors onto the centerline for sub-node extraction, but they did not provide details on how to calculate the centerline of the hallway.

In this study, we used PCA [36] to determine the main direction of the hallway. As shown in Figure 7, the red arrow indicates the main direction, which corresponds to the primary axis and aligns with the longer direction of the hallway. The yellow arrow represents the secondary direction, which is perpendicular to the primary axis. By identifying the primary axis of variance in the point cloud data, we projected the points onto this axis to define the central line. However, we observed that the centerline extracted directly using PCA occasionally drifted towards one side of the hallway. This drift occurs mainly due to the asymmetric or irregular distribution of the point cloud, as the direction of maximum variance calculated by PCA may not always perfectly align with the geometric center of the hallway. To optimize the extraction results, we introduced the centroid of the hallway as a reference, calculated by Equation (1).

Let the primary direction vector obtained from PCA be denoted as $\vec{d}$, and the centroid of the hallway as $\vec{c}$; the minimum and maximum projection values along the primary direction are represented by $proj_{min}$ and $proj_{max}$, respectively. Without correction, the endpoints derived from the projections can be expressed as

$$\vec{p}_{min} = proj_{min} \cdot \vec{d}, \quad \vec{p}_{max} = proj_{max} \cdot \vec{d} \quad (3)$$

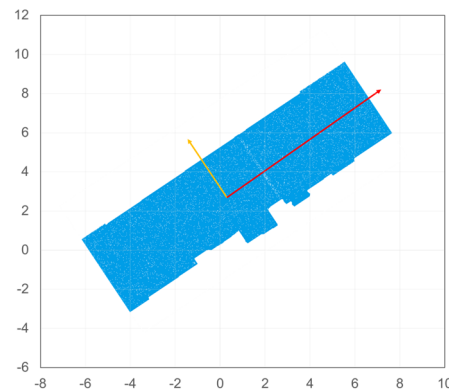


Figure 7. Principal component analysis (PCA) of the hallway point cloud: red arrow for the main direction, yellow for the secondary.

However, such endpoints may deviate from the true center of the hallway. Therefore, we correct the endpoints by considering the centroid and adjusting the positions symmetrically around it. The corrected endpoints are calculated using the following equations:

$$\begin{aligned}\vec{p}_{min} &= \vec{c} + \vec{d} \cdot (\text{proj}_{min} - \vec{c} \cdot \vec{d}) \\ \vec{p}_{max} &= \vec{c} + \vec{d} \cdot (\text{proj}_{max} - \vec{c} \cdot \vec{d})\end{aligned}\quad (4)$$

In Equation (4), $\vec{c} \cdot \vec{d}$ represents the projection of the centroid onto the primary direction, ensuring that the endpoint correction is made relative to the position of the centroid. $\vec{d} \cdot (\text{proj}_{min} - \vec{c} \cdot \vec{d})$ and $\vec{d} \cdot (\text{proj}_{max} - \vec{c} \cdot \vec{d})$ compute the offsets of the minimum and maximum projection values relative to the centroid's projection. The final endpoints $\vec{p}_{min}$ and $\vec{p}_{max}$ are determined by adding the corrected offsets to the centroid $\vec{c}$, ensuring symmetrical positioning around it.

By projecting the points onto the main direction identified by PCA and then adjusting these projections relative to the centroid, this adjustment reduced the drift of the extracted centerline, ensuring the centerline more accurately represents the true central axis of the hallway.

Subsequently, the centers of doors parallel to the centerline of each hallway are projected onto the centerline to obtain intersection points, as shown in Figure 8. These intersection points are then sorted, and if the distances between any two intersection points are less than a set threshold of 1 m, they are merged into a new intersection point. This process continues until the distances between all intersection points are greater than the set threshold, ensuring that the geometric relationships within the hallway are accurately reflected.

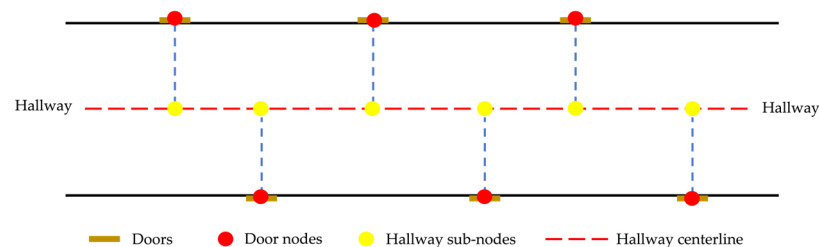


Figure 8. An illustration of multiple sub-node extraction for longer hallways.

Based on our prior knowledge, not all hallways require sub-node extraction. To avoid unnecessary sub-node extraction and improve the efficiency of the algorithm, this study classifies hallway network extraction into two types based on the length of the hallways. For hallways with a length less than or equal to 4 m, we extracted only the centroid as the

representative node. However, for hallways with a length longer than 4 m, we applied sub-node extraction to enhance the accuracy of the navigation network.

In this way, a short hallway is represented by a single node, which is connected to its associated door nodes to form the corresponding edges. In contrast, a longer hallway is represented by multiple sub-nodes within the navigation network, with each sub-node connected to either adjacent doors or neighboring sub-nodes, thereby forming the edges of the network.

2.3.3. Node Duplicate Detection for Doors

According to the file tree structure shown in Figure 5, point cloud files representing the same door can appear simultaneously in both general space (e.g., office) and transition space (e.g., hallway) directories. This setup introduces the potential issue of duplicate node extraction when processing door data, as a newly extracted door node may have already been processed and saved in the navigation network. To address this issue, we used a node duplicate detection algorithm that filters out duplicate nodes extracted from the door point cloud data.

Since the semantically segmented point cloud files representing the same door in the general space (e.g., office) and transition space (e.g., hallway) directories may have slight differences in detail, but are generally consistent, such as points distribution or number of points, the extracted centroids for the same door may be different. To accurately identify duplicate nodes, we employed a distance threshold-based method.

To ensure that each node uniquely represents a specific door, for each new door point cloud file, we calculate its centroid coordinates $C_{new} = (x_{new}, y_{new}, z_{new})$. For each existing door node in the extracted navigation network, we directly iterate through the centroid coordinates $C_e = (x_e, y_e, z_e)$ of each door, then calculate the Euclidean distance d between the new door centroid and each door centroid using the following equation:

$$d = \sqrt{(x_{new} - x_e)^2 + (y_{new} - y_e)^2 + (z_{new} - z_e)^2} \quad (5)$$

If the calculated distance d is less than the threshold, the nodes are considered duplicates and only the existing node is retained. Based on our prior knowledge, we set the threshold to 1 m. By using this threshold range, we can identify whether nodes are duplicates. This method ensures that each node is unique and accurately represents a door location within the indoor space.

2.4. JSON-Encoded Navigation Network

The final extracted data structure is saved in JSON [30] format, as illustrated in Figure 9. JSON is a lightweight data exchange format that is easy for people to read and write, and easy for machines to parse and generate. This format facilitates easy manipulation and integration with various applications, making JSON an ideal choice for data exchange, especially in web applications.

In this work, the designed JSON format data structure includes two main components: nodes and edges, as shown in Figure 9. Each node in the JSON structure represents a specific point (e.g., office, hallway, door, or staircase) in the indoor navigation network. The attributes of each node include the following:

- ID: A unique identifier for each node.
- Name: The name assigned to the node (e.g., “door”, “staircase”).
- Navigable Type: The type of space the node represents (e.g., general space, connection space, or transition space).
- Affiliation: The specific floor to which the node belongs in a multi-floor building.

- Coordinates: The x, y, and z coordinates that specify the node's position in the 3D space.
- Accessibility: A Boolean attribute indicating whether the node is wheelchair accessible.

Edges define the connections between nodes, forming the basis of the indoor navigation network. Each edge includes the following:

- From: The ID of the starting node.
- To: The ID of the ending node.

This approach enhances the interoperability of the extracted indoor navigation network data, allowing it to be seamlessly used in further computational analyses and real-world navigation applications.

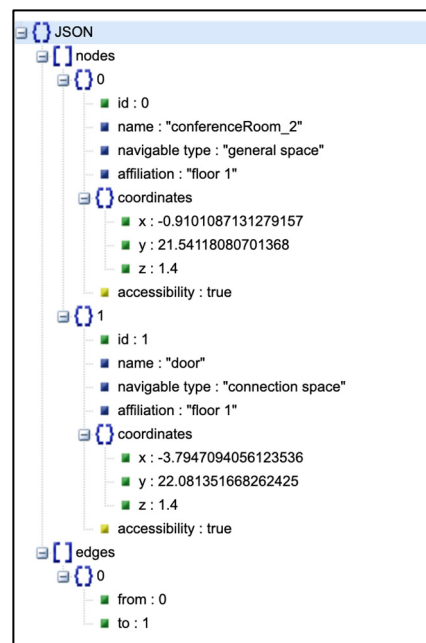


Figure 9. An example of extracted data structure in JSON format.

3. Experiments and Results

To validate the effectiveness and robustness of both our pipeline and algorithm for constructing the navigation network model, we conducted a series of experiments using the S3DIS dataset [32] and our self-collected dataset with HoloLens 2. First, we introduced the dataset used for the experiments in Section 3.1. Subsequently, in Section 3.2, we visualized and checked the distribution of nodes and edges in the extracted navigation network. Finally, in Section 3.3, we used Dijkstra's algorithm [37] to verify the shortest path planning capabilities of the extracted navigation networks.

3.1. Study Datasets

In this section, we provide a brief introduction to the public S3DIS dataset and the data we collected. Both datasets were preprocessed as described in Section 2.1. These two datasets cover a diverse range of indoor spaces, from single-floor layouts to multi-floor environments, enabling us to evaluate our method under different conditions. The following is a brief overview of each dataset.

3.1.1. S3DIS Dataset

The S3DIS dataset is an important resource for indoor scene understanding and 3D modeling studies. The dataset consists of detailed 3D point clouds data collected from

six large single-floor indoor areas within three different buildings, covering a total area of over 6000 m². These buildings cover a variety of functional spaces such as offices, conference rooms, hallways, and lobbies. The dataset also provides annotated semantic information on elements such as floors, ceilings, walls, beams, columns, windows, doors, and furniture. These richly annotated data are well suited for the development and evaluation of algorithms for automatically extracting indoor navigation networks from 3D semantic point clouds of large-scale indoor scenes. In this study, we chose Area_1, Area_4, and Area_6 in the S3DIS dataset for our experiments, as shown in Figure 10.

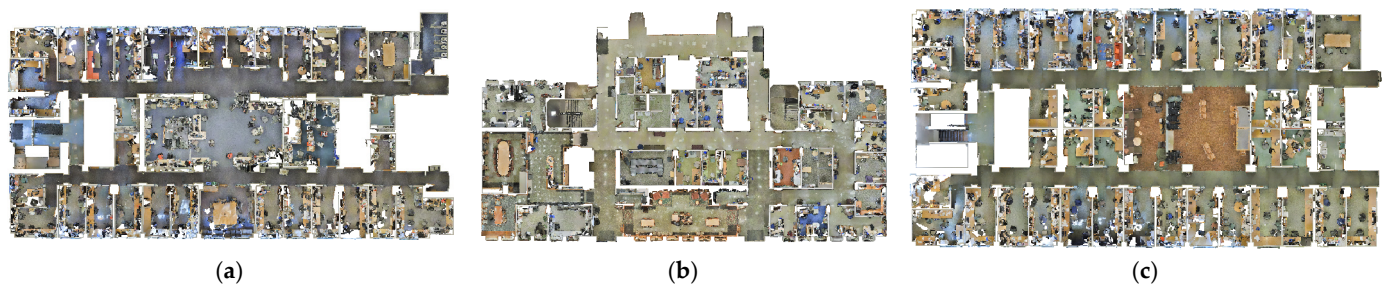


Figure 10. The three areas of the S3DIS dataset used in this study: (a) Area_1, (b) Area_4, and (c) Area_6. All images are screenshots from the original dataset and the ceilings are removed for a better visualization.

It should be noted that the S3DIS dataset does not include elevator lobbies in its space category; after data checking, we did not find where the elevator lobby was located either. Additionally, the dataset does not separately classify stairwells, instead grouping them together with hallways under the hallway category. To ensure the consistency of space classification, we added a separate category for stairwells. According to the data preprocessing strategies in Section 2.1, we preprocessed the S3DIS dataset, which mainly includes the re-segmentation of some hallways and the supplementation of virtual doors.

3.1.2. Self-Collected Dataset

Although the S3DIS dataset allows us to evaluate the performance of our method in extracting indoor navigation networks on a large scale, its scenarios are limited to single-floor environments. To address this limitation and better validate our algorithm's performance in network extraction and planning within multi-floor environments, we specifically collected a 3D dataset focused on multi-floor space.

In this study, we used the HoloLens 2 to acquire 3D data from a multi-floor building named L501. HoloLens 2 is a mixed reality headset developed by Microsoft, and several studies [15] have demonstrated its good capability for indoor mapping. Using our self-collected data, the main goal is to evaluate the algorithm's capability in extracting indoor navigation networks that connect different floors via staircases and elevators.

Specifically, we conducted data collection and registration floor-by-floor for levels 0 to 4 of the building. Our self-collected data with the scanning trajectory is shown in Figure 11.

The basic indoor units in our self-collected data include stairwells, elevator lobbies, hallways, and kitchens. Using CloudCompare, we labeled the basic indoor units on each floor with different colors: stairwells in orange, elevator lobbies in yellow, hallways in blue, and kitchens in green, as shown in Figure 12.

Subsequently, we extracted the navigation elements for each indoor unit according to the rules described in Section 2.1. As shown in Figure 13, using floor 1 as an example, in our self-collected dataset, these elements include doors, virtual doors, staircases, and elevators. Finally, we obtained the semantic point cloud data for navigation network extraction.

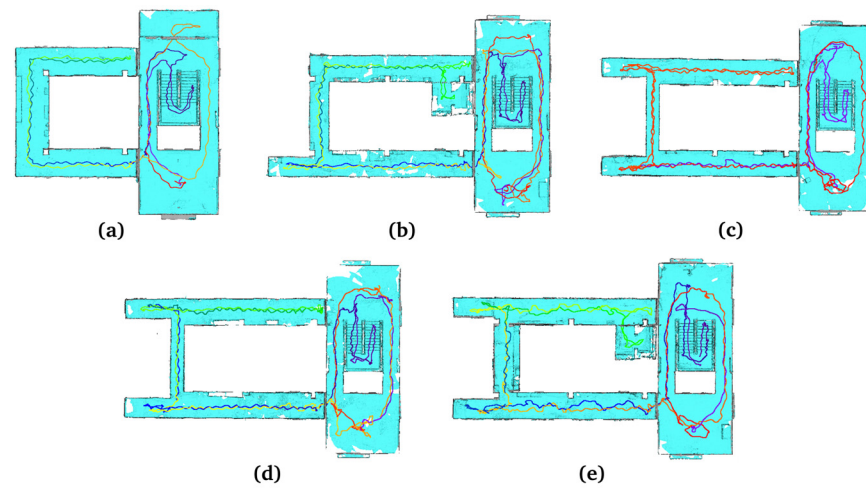


Figure 11. The self-collected 3D data and its corresponding HoloLens 2 movement trajectory: (a) floor 0, (b) floor 1, (c) floor 2, (d) floor 3, and (e) floor 4. The trajectory starts at red and ends at blue.

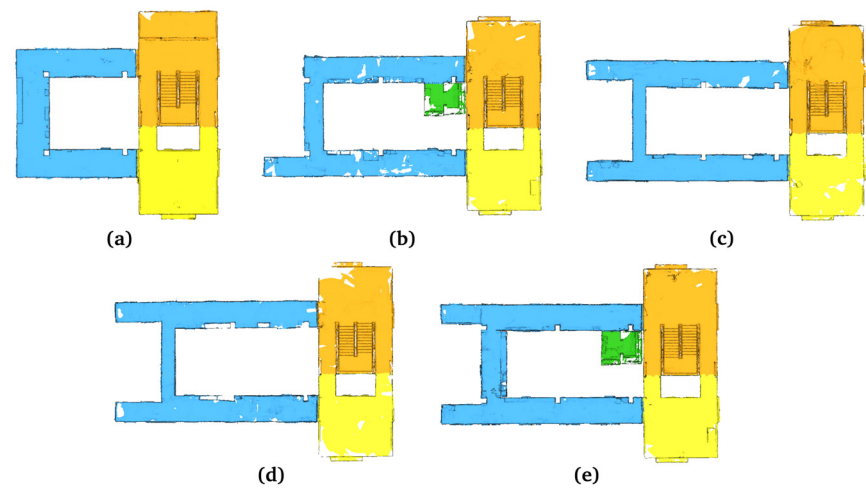


Figure 12. Color-labeled visualization of basic indoor units of our self-collected data: (a) floor 0, (b) floor 1, (c) floor 2, (d) floor 3, and (e) floor 4. In this case, stairwells are depicted in orange, elevator lobbies in yellow, hallways in blue, and kitchens in green.

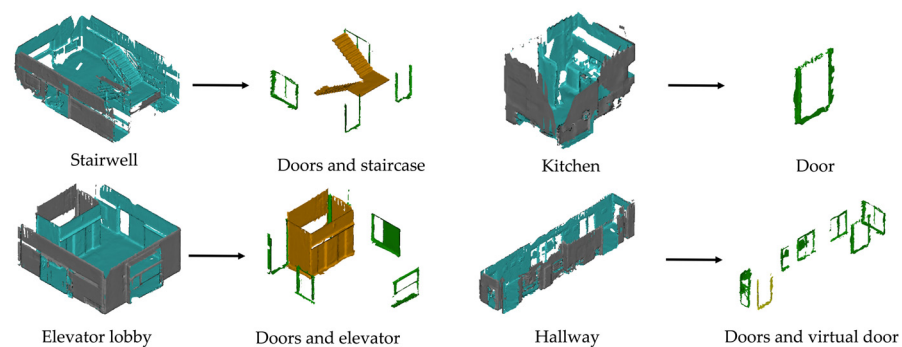


Figure 13. Indoor navigation elements were extracted from various spaces in our self-collected dataset, with floor 1 as an example.

3.2. Results of Indoor Navigation Network Extraction

The results of the extracted indoor navigation network from the experiments are shown in Figures 14 and 15. In the extracted indoor navigation network, blue node represents general space such as office, conference room, and storage; red node represents connection space such as door and virtual door; and yellow node represents transition space such as hallway and stairwell. The edges connecting the nodes are shown in green.

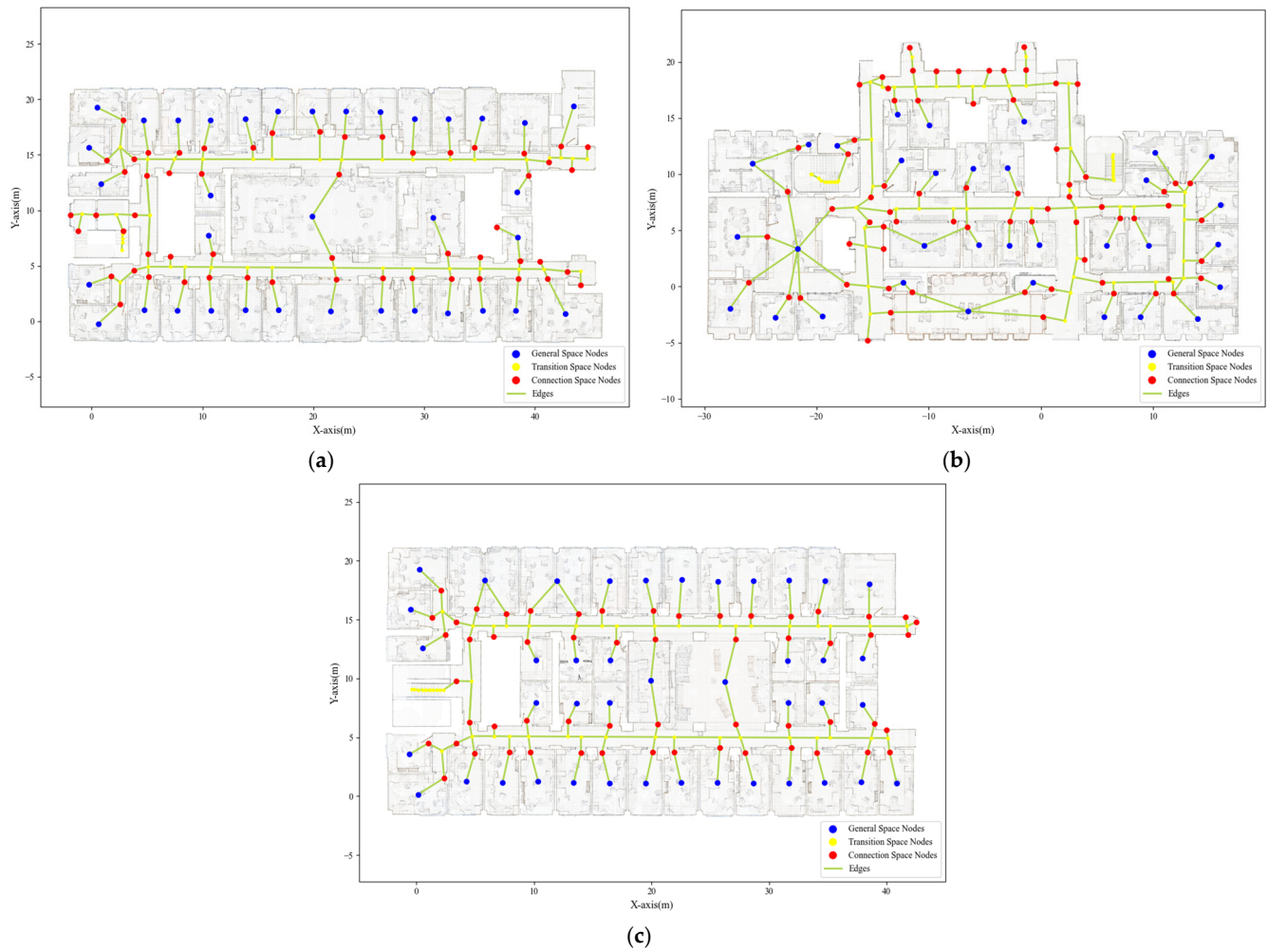


Figure 14. The results of the extracted indoor navigation network with S3DIS dataset: (a) Area_1, (b) Area_4, and (c) Area_6.

To verify that the distribution of nodes and edges corresponds to the actual spatial locations, we used point cloud data as a background map and overlaid the extracted navigation network onto it for visualization. This way provides an intuitive way to check if the extracted navigation network aligns with the actual spatial layout.

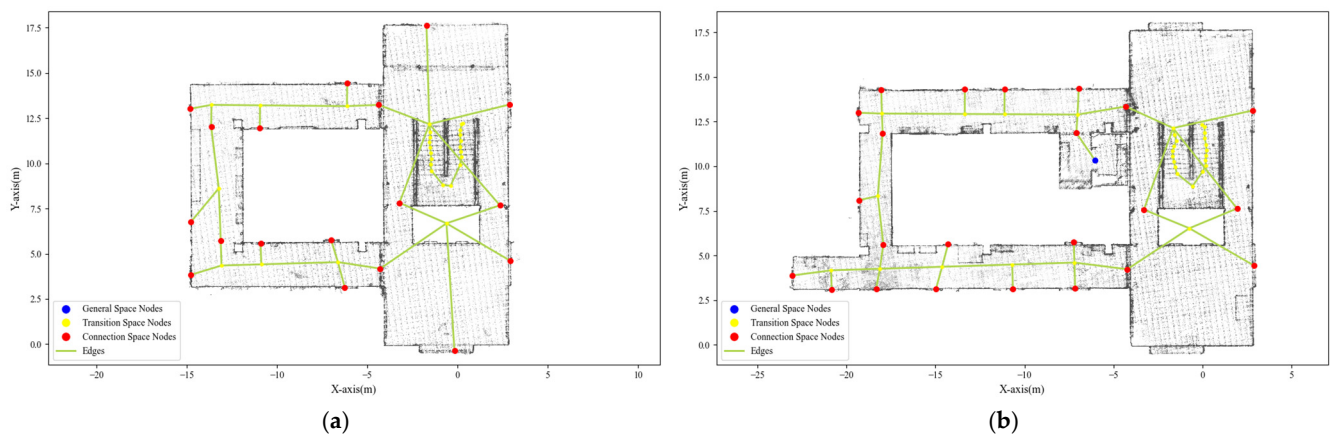


Figure 15. Cont.



Figure 15. The results of the extracted indoor navigation network with self-collected dataset: (a) floor 0, (b) floor 1, (c) floor 2, and (d) floor 3, (e) floor 4.

Additionally, to validate the efficiency of the indoor navigation network extraction algorithm, we recorded the runtime for extracting navigation networks across different datasets. For each scene, we performed the network extraction algorithm five times and took the average runtime as the evaluation reference. Moreover, we recorded the total input file size and output JSON file size for each scene. All navigation network extractions were conducted on a standard macOS laptop with a 2.3 GHz Quad-Core Intel Core i5 processor and 8 GB of RAM. The statistics are shown in Table 2.

Table 2. Input file size, output file size, and runtime statistics for navigation network extraction across different datasets.

Datasets	Scene	Input File Size (MB)	Output File Size (KB)	Average Runtime * (Seconds)
S3DIS	Area_1	3534.0	53	344.0
	Area_4	2795.5	63	355.1
	Area_6	2693.1	57	322.0
L501	F0	13.7	19	3.9
	F1	19.0	23	4.2
	F2	15.0	24	3.7
	F3	15.9	25	4.0
	F4	21.4	26	4.8

* The average runtime across the five executions.

3.3. Shortest Path Planning

To verify the spatial analysis capability of the extracted indoor navigation network, we applied the Dijkstra algorithm [37] for the shortest path planning computation with

each generated navigation network. By inputting the IDs of the start and end nodes, along with a parameter indicating whether wheelchair access is required, we can compute the shortest path connecting the start and end points within the network. This method not only verifies the accuracy of the navigation network, but also evaluates its potential in real-world applications, particularly for accessible path planning for people with mobility impairments.

Figure 16 shows the shortest path planning results for the single floor navigation network extracted from the S3DIS dataset. For the multi-floor navigation networks in the self-collected dataset, the shortest path planning results across floors are shown in Figure 17. In these figures, the start and end points are clearly marked, and the path planning results are displayed, starting from purple and ending in red, to visually present the start and end of the path.

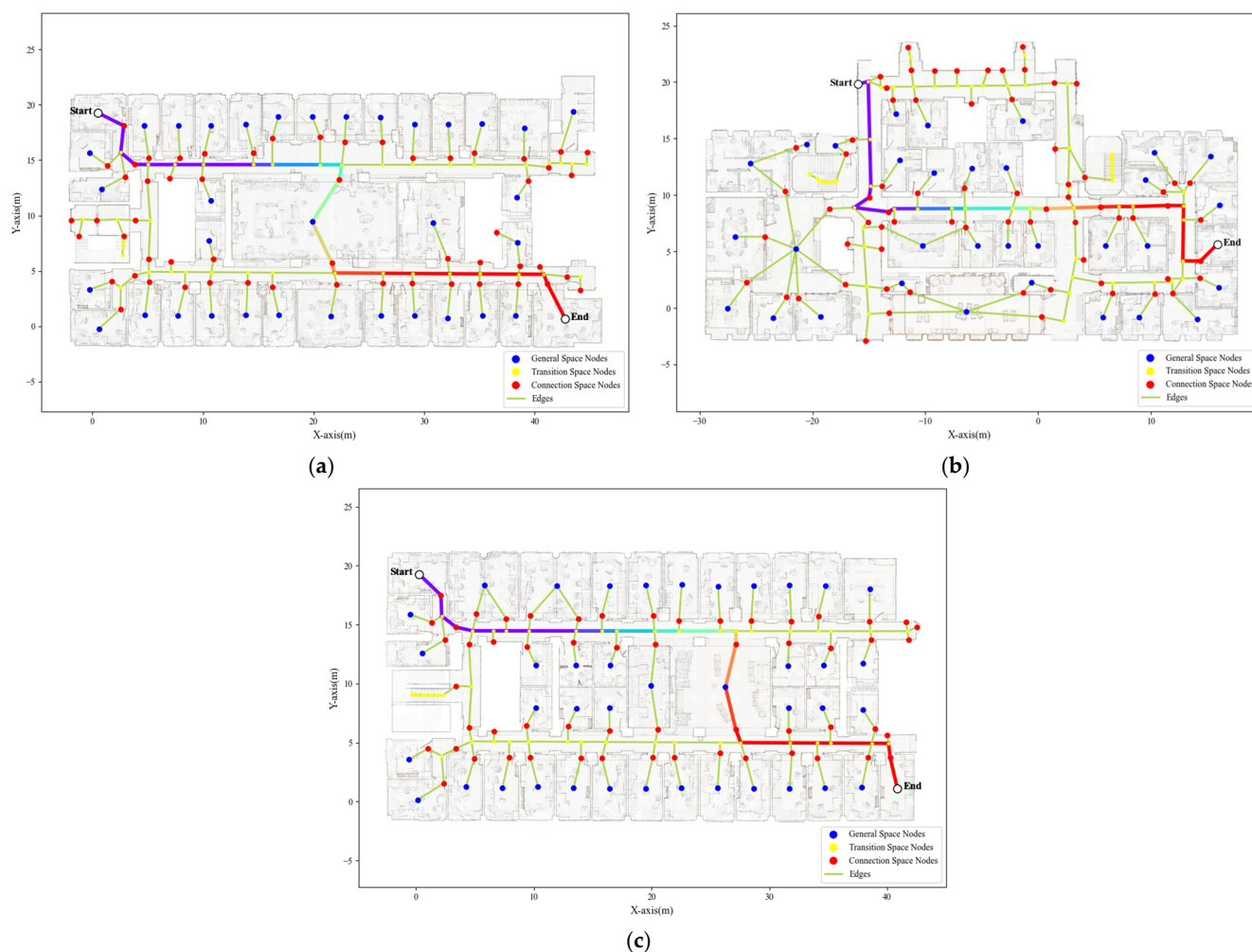


Figure 16. The results of the shortest path planning for the single floor navigation network extracted from the S3DIS dataset: (a) Area_1; (b) Area_4; (c) Area_6. The route starts at purple and ends at red.

To validate the wheelchair-accessible path planning capability, we conducted experiments on the self-collected dataset. We selected the door node at the entrance on the ground floor as the starting point and the door at the end of the hallway on the fourth floor as the end point. Due to scene constraints, the shortest path always includes the elevator, as shown in Figure 17a. To verify the effectiveness of the accessibility parameter settings, we assumed that a door node leading into the hallway on the fourth floor is not wheelchair accessible, changing its accessibility from true to false, as marked in Figure 17a,b. In this case, the wheelchair-accessible path planning results are shown in Figure 17b, where the

path was rerouted to another accessible route on the fourth floor to reach the end point. This test demonstrates the effectiveness of our extracted indoor navigation network in wheelchair-accessible path planning. Figure 17c shows the shortest path planning result when the condition is limited to staircase use only.

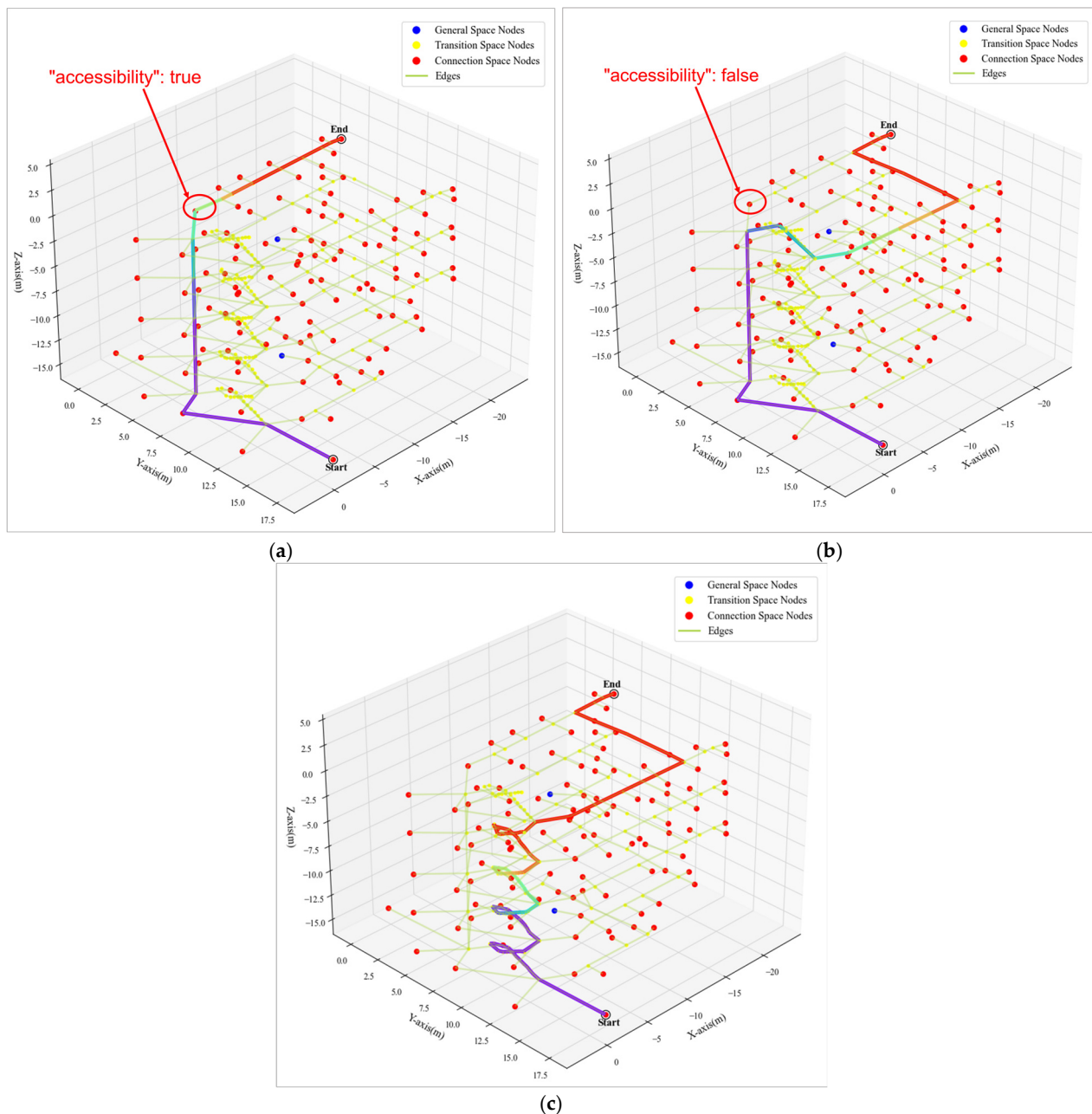


Figure 17. The results of the shortest path planning across multi-floor for the navigation network extracted from our self-collected dataset: (a) without barrier-free access limitation; (b) with barrier-free access limitation; (c) staircase-only route. The route starts at purple and ends at red.

4. Discussion

In this section, we analyzed and discussed the performance of the algorithm for indoor navigation network extraction, focusing on the robustness and efficiency of the network extraction process. In addition, through examining the shortest path planning in single-floor and multi-floor, we analyzed the path planning capability of the extracted navigation network.

4.1. Analysis of Network Extraction Robustness and Efficiency

From the visualization results in Figures 14 and 15, we can see that the distributions of nodes and edges in the extracted navigation networks are highly consistent with the actual spatial layout for different datasets. In particular, for the stairwell spaces in each scene, our network provides a more detailed representation that captures the staircase direction and layout. Compared to other studies that represent the staircase with a single node, our proposed method for automatically extracting staircase nodes results in a more refined representation that better reflects the geometric structure characteristics of the staircase.

Table 2 indicates that the runtime of our algorithm is closely related to the input data size and the complexity of the scenes. Larger input files generally require longer processing times. For smaller scenes, the algorithm efficiently extracts the navigation network within a few seconds, whereas for larger and more complex scenes, such as those in the S3DIS dataset, the runtime increases significantly. Despite the considerable differences in input file sizes, the extracted navigation network files remain lightweight, with sizes in the kilobyte (KB) range. This suggests that the extracted networks are efficiently compact.

Overall, our algorithm demonstrates good robustness and efficiency across the different indoor datasets used in this study, but there is potential for further refinement. Specifically, we have refined the extraction of staircase nodes within our self-collected dataset. As shown in Figure 15, when extracting the indoor navigation networks for stairwells and elevator lobbies, the edges connecting door nodes, elevator nodes, and staircase nodes accurately represent the topological relationships among these elements. However, due to the influence of the actual spatial structure, there are still some overlaps with the actual geometric layout. This indicates that the indoor geometric layout can still affect the accuracy of our final navigation network extraction. In practical applications, additional post-processing may be required to further adjust and optimize the extracted network to achieve a more precise indoor navigation network.

4.2. Performance in Shortest Path Planning

In single-floor environments, as illustrated in Figure 16, the navigation networks extracted from the S3DIS dataset showed robust performance, consistently generating accurate and optimal paths that followed the expected routes. The computed paths moved smoothly through different spaces like hallways, doors, and offices, proving the reliability of the network for basic indoor navigation.

For multi-floor environments, as demonstrated by the result from our self-collected dataset in Figure 17, the networks also performed well. Based on our extracted networks, the Dijkstra algorithm [37] successfully computed paths across different floors. Especially in our self-collected dataset, even after adding restrictions such as wheelchair accessibility or staircase-only, the extracted navigation network can still correctly plan the shortest paths that meet these conditions. This further validates the flexibility of our extracted navigation network in shortest path planning.

Overall, the extracted indoor navigation networks performed a robust shortest path planning capability in both single-floor and multi-floor settings. The results highlight the accuracy and utility of the networks for real-world indoor navigation, including the wheelchair-accessible path planning.

5. Conclusions

This study presents a comprehensive method for extracting accurate and practical indoor navigation network models from 3D semantic point clouds, with a focus on addressing key challenges in complex environments, such as staircases and multi-floor spaces. To ensure consistency in the data input, we developed a series of preprocessing rules. Addition-

ally, we refined the staircase node extraction algorithm and provided a more comprehensive solution for hallway sub-node extraction. Furthermore, we designed a lightweight JSON data structure to store the extracted indoor navigation network data, enabling the successful construction of indoor navigation networks using semantic point clouds.

To evaluate the effectiveness of our approach, we first validated the method using the S3DIS dataset. Subsequently, we conducted further validation on a self-collected dataset, which consisted of 3D data from a multi-floor building captured using HoloLens 2 and pre-processed to generate the required semantic point clouds. This two-step evaluation demonstrated the robustness of our method across various spatial environments. Finally, we evaluated the path planning capabilities of the navigation network using the Dijkstra algorithm. The results showed that our network effectively supports spatial connectivity analysis and performs well in different indoor environments. Additionally, wheelchair accessibility was considered, and the paths were correctly determined.

While our method successfully extracted indoor navigation networks from semantic point clouds, and applied them to shortest path planning, our research mainly focuses on Manhattan-world models [38] and static point cloud data. In the future, we aim to further improve our method by adapting it to more complex indoor environments. Additionally, we plan to integrate scene recognition and self-adaptive algorithms to achieve more efficient and flexible construction of indoor navigation networks.

Author Contributions: Conceptualization, J.H., P.H. and D.I.; methodology, J.H.; software, J.H. and P.H.; hardware, D.I. and P.H.; validation, J.H.; formal analysis, J.H. and P.H.; investigation, J.H. and P.H.; resources, D.I.; data acquisition, J.H.; writing—original draft preparation, J.H.; writing—review and editing, P.H., D.I. and J.H.; visualization, J.H. and P.H.; supervision, D.I. and P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the China Scholarship Council, Grant/Award Number: 202108130064.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the reported results can be obtained by contacting the first author or corresponding author via email.

Acknowledgments: We would like to acknowledge the Open Access Publishing Fund of the Technical University of Darmstadt for its support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Farahsari, P.S.; Farahzadi, A.; Rezazadeh, J.; Bagheri, A. A Survey on Indoor Positioning Systems for IoT-Based Applications. *IEEE Internet Things J.* **2022**, *9*, 7680–7699. [[CrossRef](#)]
2. El-Sheimy, N.; Li, Y. Indoor Navigation: State of the Art and Future Trends. *Satell. Navig.* **2021**, *2*, 7. [[CrossRef](#)]
3. Lu, M.; Arikawa, M.; Oba, K.; Ishikawa, K.; Jin, Y.; Utsumi, T.; Sato, R. Indoor AR Navigation Framework Based on Geofencing and Image-Tracking with Accumulated Error Correction. *Appl. Sci.* **2024**, *14*, 4262. [[CrossRef](#)]
4. Pang, Y.; Zhang, C.; Zhou, L.; Lin, B.; Lv, G. Extracting Indoor Space Information in Complex Building Environments. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 321. [[CrossRef](#)]
5. Liu, J.; Luo, J.; Hou, J.; Wen, D.; Feng, G.; Zhang, X. A BIM Based Hybrid 3D Indoor Map Model for Indoor Positioning and Navigation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 747. [[CrossRef](#)]
6. Lee, J. A Spatial Access-Oriented Implementation of a 3-D GIS Topological Data Model for Urban Entities. *Geoinformatica* **2004**, *8*, 237–264. [[CrossRef](#)]
7. Yan, J.; Zlatanova, S.; Diakit , A. A Unified 3D Space-Based Navigation Model for Seamless Navigation in Indoor and Outdoor. *Int. J. Digit. Earth* **2021**, *14*, 985–1003. [[CrossRef](#)]

8. Marciniak, J.B.; Wiktorzak, B. Automatic Generation of Guidance for Indoor Navigation at Metro Stations. *Appl. Sci.* **2024**, *14*, 10252. [\[CrossRef\]](#)
9. Zhang, W.; Wang, Y.; Zhou, X. Automatic Generation of 3D Indoor Navigation Networks from Building Information Modeling Data Using Image Thinning. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 231. [\[CrossRef\]](#)
10. Gorte, B.; Zlatanova, S.; Fadli, F. Navigation in Indoor Voxel Models. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*, 279–283. [\[CrossRef\]](#)
11. Zhang, J.; Wang, W.; Qi, X.; Liao, Z. Social and Robust Navigation for Indoor Robots Based on Object Semantic Grid and Topological Map. *Appl. Sci.* **2020**, *10*, 8991. [\[CrossRef\]](#)
12. Liu, L.; Li, B.; Zlatanova, S.; van Oosterom, P. Indoor Navigation Supported by the Industry Foundation Classes (IFC): A Survey. *Autom. Constr.* **2021**, *121*, 103436. [\[CrossRef\]](#)
13. Yuan, Z.; Li, Y.; Tang, S.; Li, M.; Guo, R.; Wang, W. A Survey on Indoor 3D Modeling and Applications via RGB-D Devices. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 815–826. [\[CrossRef\]](#)
14. Sun, Y.; Zhang, X.; Miao, Y. A Review of Point Cloud Segmentation for Understanding 3D Indoor Scenes. *Vis. Intell.* **2024**, *2*, 14. [\[CrossRef\]](#)
15. Hou, J.; Hübner, P.; Schmidt, J.; Iwaszczuk, D. Indoor Mapping with Entertainment Devices: Evaluating the Impact of Different Mapping Strategies for Microsoft HoloLens 2 and Apple iPhone 14 Pro. *Sensors* **2024**, *24*, 1062. [\[CrossRef\]](#)
16. Teo, T.A.; Yang, C.C. Evaluating the Accuracy and Quality of an iPad Pro's Built-in Lidar for 3D Indoor Mapping. *Dev. Built Environ.* **2023**, *14*, 100169. [\[CrossRef\]](#)
17. Hübner, P.; Landgraf, S.; Weinmann, M.; Wursthorn, S. Evaluation of the Microsoft HoloLens for the Mapping of Indoor Building Environments. In Proceedings of the 39th Annual Scientific and Technical Conference of the DGPF—Tri-Country Conference OVG—DGPF—SGPF—Photogrammetry—Remote Sensing—Geoinformation, Vienna, Austria, 20–22 February 2019; Volume 28, pp. 44–53.
18. Hübner, P.; Clintworth, K.; Liu, Q.; Weinmann, M.; Wursthorn, S. Evaluation of Hololens Tracking and Depth Sensing for Indoor Mapping Applications. *Sensors* **2020**, *20*, 1021. [\[CrossRef\]](#)
19. Weinmann, M.; Jäger, M.A.; Wursthorn, S.; Jutzi, B.; Weinmann, M.; Hübner, P. 3D Indoor Mapping with the Microsoft Hololens: Qualitative and Quantitative Evaluation by Means of Geometric Features. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *5*, 165–172. [\[CrossRef\]](#)
20. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136. [\[CrossRef\]](#)
21. Campos, C.; Elvira, R.; Rodriguez, J.J.G.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [\[CrossRef\]](#)
22. Chen, C.; Yang, B.; Song, S.; Tian, M.; Li, J.; Dai, W.; Fang, L. Calibrate Multiple Consumer RGB-D Cameras for Low-Cost and Efficient 3D Indoor Mapping. *Remote Sens.* **2018**, *10*, 328. [\[CrossRef\]](#)
23. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [\[CrossRef\]](#)
24. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5100–5109.
25. Zhang, R.; Li, G.; Wunderlich, T.; Wang, L. A Survey on Deep Learning-Based Precise Boundary Recovery of Semantic Segmentation for Images and Point Clouds. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *102*, 102411. [\[CrossRef\]](#)
26. Yang, J.; Kang, Z.; Zeng, L.; Hope Akwensi, P.; Sester, M. Semantics-Guided Reconstruction of Indoor Navigation Elements from 3D Colorized Points. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 238–261. [\[CrossRef\]](#)
27. Teo, T.A.; Cho, K.H. BIM-Oriented Indoor Network Model for Indoor and Outdoor Combined Route Planning. *Adv. Eng. Inform.* **2016**, *30*, 268–282. [\[CrossRef\]](#)
28. Flikweert, P.; Peters, R.; Díaz-Vilariño, L.; Voûte, R.; Staats, B. Automatic Extraction of a Navigation Graph Intended for IndoorGML from an Indoor Point Cloud. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*, 271–278. [\[CrossRef\]](#)
29. Qiu, Q.; Wang, M.; Xie, Q.; Han, J.; Zhou, X. Extracting 3d Indoor Maps with Any Shape Accurately Using Building Information Modeling Data. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 700. [\[CrossRef\]](#)
30. Introducing JSON. Available online: <https://www.json.org/json-en.html> (accessed on 13 December 2024).
31. IndoorGML: OGC Standard for Indoor Spatial Information. Available online: <http://www.indoorgml.net/> (accessed on 13 December 2024).
32. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543. [\[CrossRef\]](#)

33. CloudCompare, 2.13.Alpha 2023. Available online: <https://github.com/CloudCompare/CloudCompare/releases> (accessed on 13 December 2024).
34. Munkres, J.R. *Elements of Algebraic Topology*, 1st ed.; CRC Press: Boca Raton, FL, USA, 1984; ISBN 9780429493911.
35. Shang, J.; Tang, X.; Yu, F.; Liu, F. A Semantics-Based Approach of Space Subdivision for Indoor Fine-Grained Navigation. *J. Comput. Inf. Syst.* **2015**, *11*, 3419–3430.
36. Principal Component Analysis. Available online: https://en.wikipedia.org/wiki/Principal_component_analysis (accessed on 13 December 2024).
37. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *271*, 269–271. [[CrossRef](#)]
38. Furukawa, Y.; Curless, B.; Seitz, S.M.; Szeliski, R. Manhattan-World Stereo. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2010; pp. 1422–1429. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.