3D Voxel-based Path Planning for AVs in Dynamic Complex Environments

Zhaoyu Lin[1], Zhiyong Wang[1], Tailin Gong[1]

[1] South China University of Technology, Guangzhou, China, zwang1984@scut.edu.cn

Keywords: Spatial Representation, Path Planning, Deep Reinforcement Learning, Autonomous Vehicles.

Abstract

With the rapid advancement of science and technology, autonomous driving technology has been widely applied in various fields such as the automotive industry, intelligent warehousing, and emergency rescue. Traditional 2D path planning has limitations in dealing with the complex and dynamic environments of the real world, making it difficult to meet the high precision and complex path planning requirements of autonomous vehicles. With the development of data acquisition technology, large-scale point cloud data has become easily accessible, which can better extract the three-dimensional complex of the environment. Nevertheless, the unstructured nature and vast amount of point cloud data make path planning on it extremely challenging. Voxel models can effectively compress point cloud data and provide neighboring topological structures. In view of this, we propose a voxel-based framework for the path planning of autonomous vehicles in dynamic and complex 3D environments, which involves converting point cloud data into voxels and extracting navigation spaces. Subsequently, deep reinforcement learning techniques are utilized to achieve obstacle avoidance and navigation on the voxel map. It is expected that this framework will contribute to the development of 3D path planning and enhance the utilization of point cloud data in the navigation field.

1. Introduction

In recent years, with the rapid development of computer science and information technology, autonomous driving has become a emerging technology and been applied in various fields. In the early stages, autonomous vehicles (AVs) and portable robots were primarily used in simple, static environments, characterized by straightforward spatial structures, fixed obstacles, and minimal interference factors (Guo et al., 2023; Karaman and Frazzoli, 2011).These early implementations focused on controlled settings where the complexity of navigation and decision-making was relatively low. However, as the sensor technology advances, autonomous driving technology has been increasingly employed in complex and dynamic environments, such as urban settings and intelligent warehousing environments. These environments often feature complex terrains, multi-layered free spaces, and numerous static and dynamic obstacles.Urban environments, in particular, present significant challenges due to their unpredictable nature and the need for real-time responsiveness. Consequently, developing effective methods for spatial representation and path planning in three-dimensional complex spaces has become a critical research.

One of the key technologies facilitating autonomous navigation in these environments is the use of point clouds. Point clouds can directly assign 3D coordinates to the measured objects. Compared to indirect data sources that use only 2D projection points or 1D measured distances, the use of point clouds can greatly simplify surface modeling and geometric reconstruction. Therefore, point clouds are considered the most suitable data for 3D visualization in urban scenarios for various applications (Huang et al., 2020). Taking advantages of point clouds, the reconstruction of infrastructure or buildings in urban scenes has seen widespread development. However, several key challenges remain in using raw point clouds for autonomous driving scenarios:

- Unstructured Nature: Point clouds obtained through LiDAR are inherently unstructured and consist of discrete data points. Unlike structured data formats, point clouds lack an intrinsic organization or connectivity between points, making it difficult to provide the spatial relationships and topological structures of the environment directly from the raw data.
- Uneven Density: Due to differences in data collection methods and the structural characteristics of the buildings or environments, the density of point clouds would vary significantly . For instance, areas closer to the LiDAR sensor might have a higher density of points with more precise coordinates, while more distant areas might have sparser data with lower accuracy.
- Massive Volume: During data collection in urban environments, point cloud data encompasses three-dimensional spatial coordinates, color, and other diverse information, resulting in an enormous volume of data. The vast amount of point cloud data poses significant challenges for real-time or efficient computational analysis.

In this study, we use the voxel-based method to address the issues of unstructured point cloud data and challenges in efficient computation. Considering the excellent performance of deep reinforcement learning in path planning, we propose a path planning framework for autonomous vehicles in dynamic and complex environments. The framework involves voxelizing large-scale point cloud maps, constructing a deep reinforcement learning grid model on the voxelized map, and incorporating constraints for dynamic and static obstacles to achieve path planning in complex scenarios. The rest of the paper is organized as follows: Section 2 introduces previous work that is relevant to the present study. Section 3 describes the main methods. Section 4 presents the experiments performed and their preliminary results, and section 5 provides a summary of the work.

2. Related Work

2.1 Voxel Representation in 3D Space

Grid-based methods can convert the representation of a point cloud into grid cells, with each cell storing spatial information, thereby compressing a large amount of point data. Grid-based spatial representation method can save a significant amount of memory while preserving as much spatial information as possible, making it an effective technique for sensor data representation in intelligent vehicles and robots. Herbert et al. (1989) has proposed a 2.5D grid model that adds height information to a two-dimensional grid, simplifying information along the Z-axis. However, this method may lose some critical 3D information, making it difficult to accurately represent complex terrains or obstacles. Gadelha et al. (2018) involved constructing a KD-tree for point cloud data to establish topological relationships between discrete points, enabling rapid neighborhood-based searches. However, when the point cloud is too dense, the KD-tree is hard to partition the space effectively, and once built, updating the KD-tree becomes very challenging. Hornung et al. (2013) proposed the OctMap which based on an octree structure to map three-dimensional space by dividing it into eight equal cubic parts, significantly saving storage space and supporting dynamic addition and deletion of nodes. Nevertheless, in dynamic environments, OctMap needs real-time updates as the environment changes, and the complex structure of the octree increases data access and computation costs.

Voxel models can explicitly represent the positions of discrete points and their neighboring topology, providing an informative way to compress and store data. Additionally, voxel models use a regular grid representation, which simplifies and enhances the efficiency of processing three-dimensional data. Park et al. (2010) applied the KD-tree algorithm to voxelize point cloud data, achieving 3D visualization of trees. Bitelli et al. (2016) proposed a method to generate voxel models with variable resolution from scanned point clouds for historical building modeling and structural analysis. By generating voxel models at different resolutions, their method flexibly adapts to the application needs of building analysis, which is particularly useful for dealing with the complexities of historical building structures. Wang et al. (2020) introduced a method to voxelize BIM models and create a cell-based environment for path planning.Xiong et al. (2017) used a voxel-based method to extract free multi-level indoor spaces from 3D building models described by CityGML LoD4, converting the complete extraction results into voxel models using geometric and semantic relationships. Bonczak and Kontokosta (2019) utilized voxelization to model large-scale, heterogeneous, high-density urban environments from aerial LiDAR point cloud data, facilitating the analysis of urban settings. In this study, the voxel-based approach has bee used to represent the 3D complex environment and to support path planning for UGV.

2.2 3D-Voxel based Path planning

Most of the existing research and applications of voxel navigation technology are primarily focused on indoor environments. Staats (2017) proposed a method for identifying walkable surfaces within indoor buildings using a voxel model derived from point cloud analysis and corresponding Mobile Laser Scanning (MLS) trajectories. Their approach effectively segments the indoor environment into navigable and non-navigable areas, facilitating safe and efficient indoor navigation.Fichtner et al. (2018) introduced an octree-based method for extracting navigable spaces in multi-story buildings. Their method enhances the voxel model with semantic information, which is particularly useful for multi-level indoor pedestrian navigation.Li et al. (2018) developed a method for planning general Unmanned Aerial Vehicle (UAV) paths in known indoor environments using voxel models. They utilized the A* algorithm to generate both the safest shortest path and the safest lowest-cost path for UAVs. This approach ensures that the UAV can navigate efficiently while minimizing risk.However, this method only considers static indoor environments. Compared to indoor environments, outdoor structures are often more complex and subject to more frequent and unpredictable dynamic interferences, presenting greater challenges for voxel-based navigation. Schulze (2010) constructed voxel maps in agricultural environments and performed path planning for mobile robots, but this method did not account for the interference of dynamic obstacles. Asvadi et al. (2016) proposed a voxel-based method for representing obstacles above the ground using point cloud data and an inertial navigation system, constructing a static map that distinguishes between moving parts and the environment. Soilán et al. (2018) utilized voxel-based methods to automatically extract and label road networks in urban environments. The methods mentioned above lack consideration of dynamic obstacles and are not applicable in UGV navigation in 3D complex dynamic urban envionment

2.3 DRL-based Path Planning

Deep Reinforcement Learning (DRL) has emerged as a cutting-edge approach to solving 3D path planning problems, offering substantial advancements over traditional methods. DRL leverages the powerful representation capabilities of deep neural networks to effectively manage high-dimensional state spaces, making it particularly well-suited for navigating intricate and dynamic environments. By interacting with the environment, employing trial-and-error methods, and utilizing a reward mechanism, DRL agents progressively learn optimal behaviors. These agents continually adjust and refine their strategies based on real-time feedback, enabling them to adapt to changing conditions and improve their performance over time.Wang et al. (2018) proposed a modular DRL algorithm for complex dynamic maze environments, dividing the navigation task into an obstacle avoidance module and a navigation module. The obstacle avoidance module handles dynamic obstacle information, nabling the agent to react in real-time to moving obstacles and prevent collisions. Meanwhile, the navigation module explores and searches for optimal paths. By separating these functionalities, the modular design allows the algorithm to effectively manage the complexities of dynamic environments, ensuring both safety and efficiency in navigation. Zhao et al. (2020) introduced an autonomous local path planning algorithm based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) strategy, leveraging UAV autonomous decision-making to address local obstacle avoidance and path planning in unfamiliar environments.The TD3 strategy enhances the stability and performance of the DRL agent by addressing overestimation bias in Q-learning, allows the UAV to make more accurate and reliable decisions in real-time. However, the algorithm performs poorly in the presence of dynamic obstacles. Huang et al. (2023) proposed a DRL-based path planning method that achieves a good balance between distance and energy consumption in 3D environments.By balancing these two factors, the proposed method ensures that the planned paths are both efficient and practical for long-term operations in complex environments. The application of DRL in 3D path planning shows significant potential for handling complex, dynamic, and uncertain environments. The ability of DRL to learn and adapt in real-time offers a substantial advantage over traditional path planning algorithms.

3. Methodology

In this section, we will elaborate on the comprehensive framework designed for Autonomous Vehicles operating within a 3D urban environment. The complexity of urban environment, with dynamic obstacles and intricate structures, requires a robust and adaptive approach to navigation and decision-making.. To address these challenges, our framework is divided into two components: voxel-based spatial representation and Deep Reinforcement Learning (DRL)-based path planning.

3.1 Voxel-based Spatial Representation

3.1.1 Proprocessing of Point Cloud: The preprocessing of point cloud data primarily involves spatial filtering and point cloud simplification, aiming to facilitate the subsequent transformation of the point cloud map into a voxel-based representation.

Due to some limitations such as the equipment used for data collection and environmental conditions, point cloud data may contain certain levels of noise and outliers. Failure to filter the data could significantly impact the subsequent voxelization process. Hence, spatial filtering of the point cloud data is necessary. In this study, we employed a three-dimensional statistical filtering algorithm. This algorithm involves two steps: Step 1 calculates the average distance of each point to its k nearest neighbors and determining a distance threshold equal to $\mu \pm \alpha \cdot \sigma$, where μ and σ are the mean and standard deviation of these distances. Step 2 classifies points as outliers or inliers based on whether their average neighborhood distance falls below or above this threshold. We utilized a statistical filtering algorithm with parameters α=1 and k=30 to remove outliers from the point cloud. This algorithm has been demonstrated effective in removing outliers.

Point cloud simplification refers to the process of reducing or lowering the data volume of point cloud data while preserving key information. It aims to decrease the number of points in the point cloud, thereby facilitating data processing and reducing computational costs associated with voxelization.

3.1.2 Point Cloud Semantic Segmentation : We perform semantic segmentation on the preprocessed point cloud data to extract road labels and delineate areas accessible to autonomous vehicles. We employ the PointNet++ (Qi et al., 2017), a deep learning model specifically designed for handling 3D point cloud data. PointNet++ captures the spatial structure and semantic information of point cloud data effectively through hierarchical local feature extraction. Using PointNet++, each point is classified and assigned to predefined semantic labels, such as roads, buildings, and vehicles. Only the points labeled as roads are retained, providing a foundation for subsequent voxel-based navigation space extraction and path planning.

3.1.3 Navigation Space Construction: After completing the semantic segmentation phase, we proceed to standard voxelization of the segmented point cloud data to construct the navigation space for autonomous vehicles. The point cloud data is divided into a standard voxel grid using a fixed resolution, with each voxel unit being a cube whose edge length is determined by the predefined resolution. Each point is then assigned to its corresponding voxel unit, forming a standard voxel representation.

During the process of point cloud data acquisition, some issues such as scanning procedures and occlusions caused by vehicles may result in the loss of certain portions of road point cloud data. Additionally, the preprocessing of point cloud data and voxel semantic segmentation may lead to the presence of numerous holes in the extracted navigation space, thereby affecting the subsequent path planning to some extent. To address this, hole repair is performed through directed voxel fusion, followed by convergence optimization using various partial differential equations based on the surface information of the hole neighborhood.

3.2 3D Path Planning

3.2.1 Path Presentation: In voxel maps, paths are depicted as a set of voxels with specific occupancy states. When an agent occupies specific voxels in the voxel map at the current time step, the occupancy states of these voxels are represented as a path. Voxels marked as part of the path maintain their occupancy states even after the agent has moved away. Thus, paths in voxel maps represent the historical trajectory of agent movement.

Considering the constraints of real physical laws and vehicle driving conditions, the path must satisfy the following constraints:

1. The value of all voxels in each path point must be 0. That is, the path cannot pass through obstacles. For any voxel v_i in path R, $\forall v_i \in r, v_i = 0$.
2. The voxels at the bottom of the path must be located on the ground. For path R, let the bottom voxel be $V_b = v(x_b, y_b, z_b)$, and the voxels representing the ground below be $V_g = (x_g, y_g, z_g) = (x_b, y_b, z_b - 1)$. $\forall v_i \in V_g, v_i = 1$

3.2.2 DRL model In the voxel map, the agent is represented as a cube comprising multiple voxels. We use the agent's movement trajectory within an episode to represent the path. At time t given that the occupancy state set of all voxels in the voxel map is E_t, the occupancy state of a voxel is denoted as $e(x, y, z, t)$, where x, y, z are the coordinates of the voxel in the map. In the voxel map, the occupancy state e can take the following values:

0: Unoccupied. This state indicates that the voxel is free space.

1: Obstacle. This state indicates that the voxel is an obstacle or part of an obstacle.

2: Path. This state indicates that the voxel is part of a path.

We use the set of occupation states of all voxels in the voxel map as the state space of the model.

$$s_t = [e(x, y, z, t) | e \in E_t] \tag{1}$$

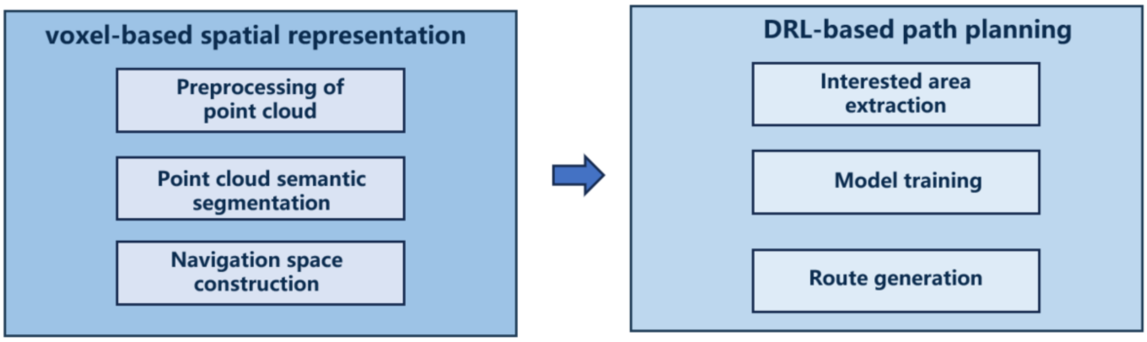

Figure 1. The study framework.

We use the change in the agent’s position incrementally as the actions.

$$a_t = [\Delta x, \Delta y, \Delta z], \Delta x, \Delta y, \Delta z \in [-1, 0, 1] \quad (2)$$

When the action $a = [\Delta x, \Delta y, \Delta z]$ is taken, the position of the agent changes from (x, y, z) to $(x + \Delta x, y + \Delta y, z + \Delta z)$.

This action space allows the agent’s range of movement as shown in Figure 2:

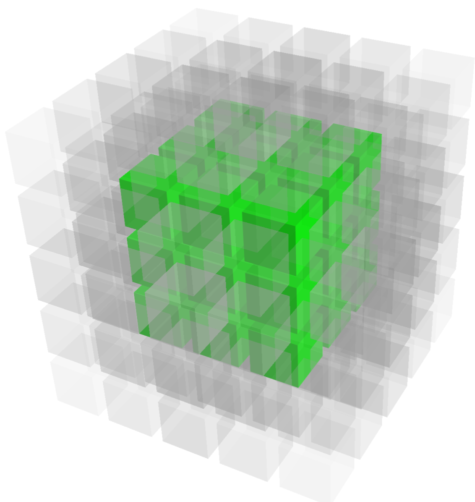

Figure 2. For an agent size of $l = 3$, the range of movement is depicted. The green areas represent the current positions of the agent, while the gray areas indicate the possible positions the agent can move towards.

In essence, this position change increment represents the direction of the agent’s movement. Therefore, we encode this increment into 26 states and transform a_t into an integer in the range 0 to 26.

The reward function is a critical component that directly impacts the agent’s learning effectiveness within the environment, guiding its actions and decisions towards achieving optimal performance. Our reward function comprises several key components, each contributing to different aspects of the agent’s navigation and decision-making process. The reward function is mathematically expressed as follow:

$$r = \alpha d_1 + \beta d_2 + C_1 + C_2 \quad (3)$$

The reward function includes the distance components αd_1 and βd_2. Here, d_1 is the distance from the agent’s current position to the starting point, and d_2 is the distance from the agent’s current position to the endpoint. α and β are the coefficients for these distances. C_1 and C_2 are constant components. C_1 is awarded when the agent reaches the endpoint, and the episode terminates. C_2 is a penalty given when the agent is located in an obstacle voxel, i.e., when the agent collides with an obstacle, causing the episode to terminate.

4. Preliminary Results

As the first step in our study, we used the point cloud dataset from the Lands Department of the Government of the Hong Kong Special Administrative Region to evaluate our spatial representation. Figure 3 is a point cloud environment data of a certain area in Kowloon, Hong Kong. It contains different objects including buildings, roads, trees and cars, offering a detailed and diverse urban landscape for our analysis. After the preprocessing, we classified the point cloud and extracted the navigable space. After that, we extracted the part with semantic labels including buildings, trees and roofs to generate navigable space point cloud data (Figure 4). With the voxel size $l = 0.5$, we voxelized the navigable space to generate the final voxel map. Figure 5 is the voxel map generated from navigable space.

5. Conclusions and Future Work

In this paper, we propose a framework that leverages voxel models and deep reinforcement learning (DRL) algorithms to achieve path planning in dynamic and complex environments. Our approach begins with semantic segmentation of point clouds, followed by voxelization to generate the navigable spaces. We then employ a DRL algorithm to perform path planning on the voxel maps, generating routes that effectively avoid obstacles. Currently, we are implementing the DRL-based path planning and are building diverse training datasets to train our model for robust performance in various settings.

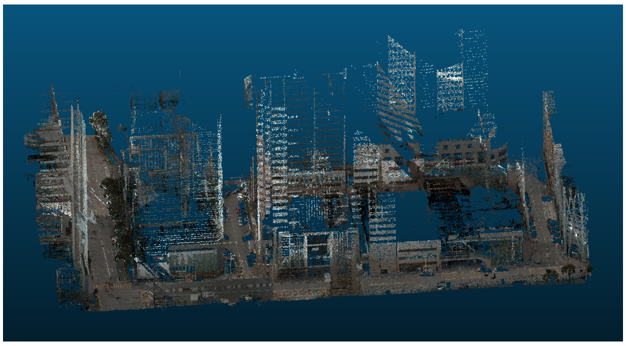

Figure 3. The original point cloud data

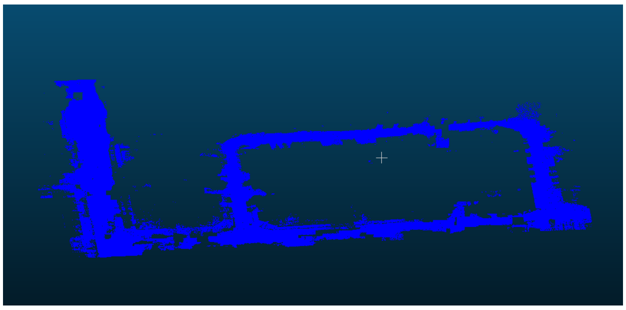

Figure 4. Road point cloud

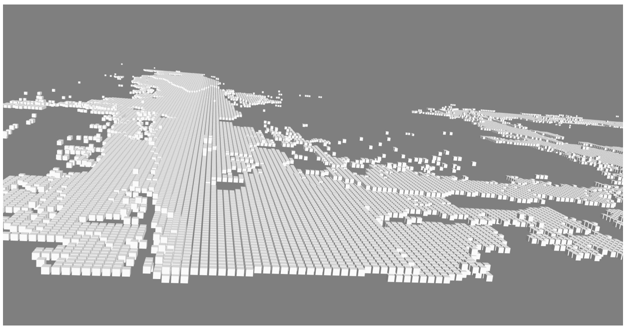

Figure 5. Voxel maps generated from road point cloud

Acknowledgement

This study is supported by "the Fundamental Research Funds for the Central Universities"(NO.2023ZYGXZR056).

References

Asvadi, A., Premebida, C., Peixoto, P., Nunes, U., 2016. 3D Lidar-based Static and Moving Obstacle Detection in Driving Environments: An Approach Based on Voxels and Multi-Region Ground Planes. *Robotics and Autonomous Systems*, 83, 299–311.

Bitelli, G., Castellazzi, G., D'Altri, A. M., De Miranda, S., Lambertini, A., Selvaggi, I., 2016. AUTOMATED VOXEL MODEL FROM POINT CLOUDS FOR STRUCTURAL ANALYSIS OF CULTURAL HERITAGE. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B5, 191–197.

Bonczak, B., Kontokosta, C. E., 2019. Large-Scale Parameterization of 3D Building Morphology in Complex Urban Landscapes Using Aerial LiDAR and City Administrative Data. *Computers, Environment and Urban Systems*, 73, 126–142.

Fichtner, F. W., Diakité, A. A., Zlatanova, S., Voûte, R., 2018. Semantic Enrichment of Octree Structured Point Clouds for Multi-Story 3D Pathfinding. *Transactions in GIS*, 22(1), 233–248.

Gadelha, M., Wang, R., Maji, S., 2018. Multiresolution Tree Networks for 3D Point Cloud Processing. 11211, Cham, 105–122.

Guo, H.-X., Hong, T.-T., Zhang, Y.-W., 2023. Path Planning Simulation and Research of Mobile Robot in Static Environment Based on Genetic Algorithm. *2023 8th International Conference on Automation, Control and Robotics Engineering (CACRE)*, 226–230.

Herbert, M., Caillas, C., Krotkov, E., Kweon, I., Kanade, T., 1989. Terrain Mapping for a Roving Planetary Explorer. *Proceedings, 1989 International Conference on Robotics and Automation*, 997–1002.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*, 34(3), 189–206.

Huang, R., Hong, D., Xu, Y., Yao, W., Stilla, U., 2020. Multi-Scale Local Context Embedding for LiDAR Point Cloud Classification. *IEEE Geoscience and Remote Sensing Letters*, 17(4), 721–725.

Huang, S., Wu, X., Huang, G., 2023. Deep Reinforcement Learning-Based Multi-objective 3D Path Planning for Vehicles. *Proceedings of 2023 Chinese Intelligent Systems Conference*, 867–875.

Karaman, S., Frazzoli, E., 2011. Sampling-Based Algorithms for Optimal Motion Planning. *The International Journal of Robotics Research*, 30(7), 846–894.

Li, F., Zlatanova, S., Koopman, M., Bai, X., Diakité, A., 2018. Universal Path Planning for an Indoor Drone. *Automation in Construction*, 95, 275–283.

Park, H., Lim, S., Trinder, J., Turner, R., 2010. Voxel-based volume modelling of individual trees using terrestrial laser scanners. *Proceedings of 15th Australasian Remote Sensing & Photogrammetry Conf., Alice Springs, Australia*, 1125–1133.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 30.

Schulze, M., 2010. 3D-camera based navigation of a mobile robot in an agricultural environment. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 538–542.

Soilán, M., Truong-Hong, L., Riveiro, B., Laefer, D., 2018. Automatic Extraction of Road Features in Urban Environments Using Dense ALS Data. *International Journal of Applied Earth Observation and Geoinformation*, 64, 226–236.

Staats, B., 2017. Identification of Walkable Space in a Voxel Model, Derived from a Point Cloud and Its Corresponding Trajectory.

Wang, Q., Zuo, W., Guo, Z., Li, Q., Mei, T., Qiao, S., 2020. BIM Voxelization Method Supporting Cell-Based Creation of a Path-Planning Environment. *Journal of Construction Engineering and Management*, 146(7), 04020080.

Wang, Y., He, H., Sun, C., 2018. Learning to Navigate Through Complex Dynamic Environment With Modular Deep Reinforcement Learning. *IEEE Transactions on Games*, 10(4), 400–412.

Xiong, Q., Zhu, Q., Du, Z., Zlatanova, S., Zhang, Y., Zhou, Y., Li, Y., 2017. Free Multi-Floor Indoor Space Extraction from Complex 3D Building Models. *Earth Science Informatics*, 10(1), 69–83.

Zhao, H., Xi, X., Wang, C., Pan, F., 2020. Ground Surface Recognition at Voxel Scale From Mobile Laser Scanning Data in Urban Environment. *IEEE Geoscience and Remote Sensing Letters*, 17(2), 317–321.