# Large Scale Point Cloud Semantic Segmentation for Indoor Digital Twin Generation

Johann Nikolai Hark[1,*], Bernd Schaeufele[2,*] and Ilja Radusch[1]

[1]*Daimler Center for Automotive Information Technology Innovations (DCAITI), Berlin, Germany*

[2]*Fraunhofer Institute for Open Communication Systems (FOKUS), Berlin, Germany*

## Abstract

Automated processing of large amounts of sensor data poses a significant challenge in many fields, particularly with LiDAR point clouds. One field of application for LiDAR is the creation of high-definition (HD) maps, which are utilized in various domains, such as mobile indoor navigation. For visually impaired individuals, indoor navigation is crucial for enhancing their quality of life and independence. In this work, we focus on two advanced methods for semantic segmentation of point clouds generated using LiDAR and 360° camera sensors. These methods generate digital twins to create accurate and detailed representations of physical environments. The digital twins can be used to produce HD maps for indoor navigation. The first method involves converting the point cloud into a graph structure known as a superpoint graph (SPG). The second method, RandLA-Net, is based on efficient random sampling of points within the point cloud. Both methods are evaluated with our dataset, achieving an overall accuracy of 89%. This reproduced performance is consistent with their results on the public point cloud benchmark from Stanford University, demonstrating the efficiency of state-of-the-art semantic segmentation methods.

## Keywords

indoor mapping, SLAM, semantic segmentation, digital twin, point clouds

## 1. Introduction

Accurate HD maps are essential for indoor navigation for visually impaired [1], but also the foundation for other applications, such as automated driving [2, 3]. Creating digital maps manually via CAD and laser measurements is a laborious task. However, advancements in 3D acquisition technologies such as LiDAR, TOF, and RGB-D cameras have made these sensors more accessible and affordable. SLAM techniques are used to convert 3D sensor data into point clouds (PCs), which form the basis of automated generation of digital twins. State-of-the-art SLAM methods can process large amounts of data in real-time and map extensive environments efficiently.

We use two indoor sensor platforms with modular hardware: two LiDAR sensors, a 360° camera, and an industrial-grade IMU. Our backpack supports 3D SLAM with six degrees of freedom (DOF), and the trolley is designed for 2D SLAM with three DOF, reducing complexity and potential mapping errors. We conduct mapping experiments in various indoor environments at the Fraunhofer Institute for Open Communication Systems (FOKUS) and the Daimler Center for Automotive Information Technology Innovations (DCAITI). The sensor data is processed with the Google Cartographer SLAM algorithm [4], requiring time synchronization and sensor calibration. The recordings contain over 40 minutes of sensor data from various physical environments, generating PCs with more than 673 million points. The data acquisition pipeline generates models of indoor environments with pointwise RGB coloring.

Subsequently, the PCs are processed for obtaining semantic labels to create a digital twin. For this, we apply semantic segmentation, i.e., classify each point into a predefined class. For the supervised semantic segmentation, the PCs are manually annotated in 11 semantic classes for training. Finally, we

*IPIN-WCAL 2025: Workshop for Computing & Advanced Localization at the Fifteenth International Conference on Indoor Positioning and Indoor Navigation, September 15–18, 2025, Tampere, Finland*

[*] Corresponding authors.

j.hark@dcaiti.com (J. N. Hark); bernd.schaeufele@fokus.fraunhofer.de (B. Schaeufele); ilja.radusch@dcaiti.com (I. Radusch)

evaluate two state-of-the-art semantic segmentation methods, achieving significant results. In particular, RandLA-Net shows an overall accuracy of 89%, demonstrating its effectiveness on the generated PCs.

The paper is structured as follows. Existing approaches for semantic segmentation of large PC data are discussed in section 2. In section 3, the hardware platform and the data recording process are described. Consequently, in section 4, the method for automated digital twin generation is explained. The results are evaluated in section 5, before a conclusion and an outlook are given in section 6.

## 2. Related Work

Creating digital twins of indoor environments is costly and time consuming, especially when updating existing plans. Mobile systems are more flexible and efficient compared to static tripod solutions such as the Matterport scanner, simplifying mapping by eliminating occlusion studies and multiple measurement positions. Current mobile indoor mapping systems, categorized by their physical configurations—handheld, backpack, and trolley [5] typically use LiDAR or RGB-D cameras. Handheld devices, like the ZEB-HORIZON and ZEB-REVO-RT [6], prioritize a lightweight design to prevent operator fatigue and often incorporate 2D LiDAR sensors. Backpack systems, although heavier, distribute weight across the user's body, allowing more robust sensors, typically LiDAR for its longer range. Examples include Google's Cartographer backpack [4] and NavVis VLX [7]. Trolley systems, the least weight-constrained, integrate high-quality sensors like terrestrial LiDAR sensors for superior mapping accuracy.

SLAM algorithms rely on available sensors, with LiDAR being common. Scan-to-scan matching with algorithms like ICP [8] matches consecutive LiDAR scans to compute relative pose changes but accumulates errors over time. Scan-to-map matching requires good initial pose estimates, mitigating error accumulation, while being robust and efficient [4]. The LOAM algorithm [9] uses LiDAR points, optionally combined with IMU data, to achieve low drift and low computational complexity. LOAM splits SLAM into high-frequency, low-fidelity odometry (scan-to-scan) and slower, high-accuracy PC registration (scan-to-map). VLOAM [10] extends LOAM by adding high-frequency monocular camera data for robustness to heavy motion and visual feature scarcity.

Particle-filter methods minimize local error accumulation but can be resource-intensive, though smaller dimensional feature representations can help [11]. Loop closure, essential for global SLAM, uses histogram-based matching [12], feature detection [13], and graph-based methods [14]. Optimization techniques minimize errors from these constraints [15], and recent approaches incorporate semantic segmentation [16]. Extensive reviews of SLAM algorithms are available [17].

Initial success in 2D image segmentation is achieved using advanced neural network architectures such as [18, 19, 20]. However, it cannot be directly applied to PCs due to its irregular, non-uniform nature and varying density, complicating the use of standard convolutional neural networks.

Thorough reviews on PC segmentation are given by Grilli et al. [21] and Lu et al. [22], categorizing methods into projection-based, discretization-based, point-based, and hybrid approaches. Projection and discretization convert PCs into regular representations like multi-view or volumetric forms, enabling the use of 2D architectures [23]. Discretization methods like SEGCloud [24] preserve neighborhood structures but can introduce artifacts and information loss.

Point-based methods, such as PointNet [25], directly process PCs and capture local geometries, e.g., RandLA-Net [26]. Point convolution methods like KPConv [27] offer efficient solutions. RNN-based methods, such as RSNet, capture context features. Graph-based methods, like the approach by Landrieu and Simonovsky [28], use graph neural networks to detect geometric structures.

In [29], an indoor semantic segmentation approach based on the FCGF architecture [30] is shown, which allows using only partially labeled PC datasets. By registering PCs through overlapping regions, labels can be transferred. UnScene3D [31] is an unsupervised method for 3D indoor LiDAR segmentation. It applies instance segmentation with a 3D Transformer architecture [32], working on RGB-D sensor data, though. The authors in [33] present Swin3D, a pretrained Transformer backbone for 3D indoor semantic segmentation, which is an extension of the Swin Transformer architecture [34]. The method is evaluated with real world data, but the model is trained on the synthetic Structured3D dataset [35].

## 3. Data Acquisition

One main design goal of the indoor mapping platforms is high-quality, comparable sensor data on different platforms. Our design is compact, lightweight, yet stiff for mounting on a backpack or a trolley. It features optimized sensor mounting positions to avoid interference and aid calibration. Mounting the camera directly under the main LiDAR ensures an unobstructed horizontal field of view. The platform, made of lightweight aluminum with 3D-printed parts, holds two LiDARs, an industrial-grade IMU, and a dual-lens fish-eye camera. The sensors and sampling rates are listed in Table 1.

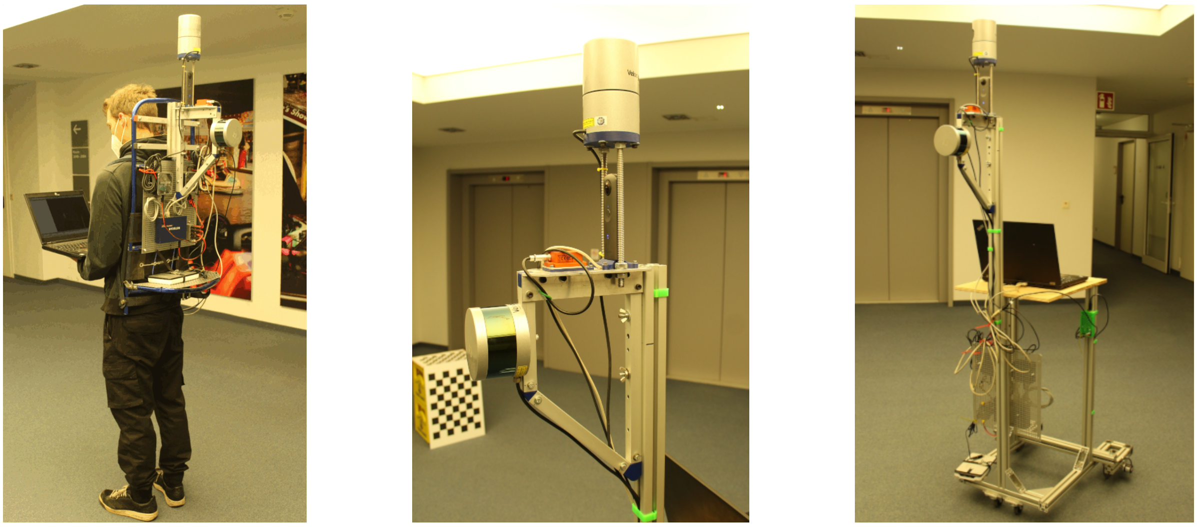
The different sensor setups are shown in Fig. 1. The backpack platform (Fig. 1a) is lightweight yet stiff, allowing the operator to move freely while mapping larger environments. The sensor platform (Fig. 1b) can be adjusted for different heights of the operator. The trolley (Fig. 1c) reduces DOF in motion to simplify the underlying SLAM problem. Built from aluminum extrusions, it features a sensor platform and a table top for a notebook. Wheel-encoders are installed for additional mapping accuracy, as shown in Fig. 2b, making the trolley a differential drive robot.

Camera and LiDAR calibration is essential [36], and calibration, including various cameras and LiDAR sensors, is complex and time-consuming. As all sensors needing calibration are mounted on the same platform, calibration is required only once. The camera software includes intrinsic parameters, requiring extrinsic calibration only. The sensor platform is mounted on the trolley and placed in a static environment with various test objects (Fig. 2a), including chess boards and colored objects at different distances and angles. Single shots are recorded with the trolley from different locations.

Recordings are used for manual calibration. LiDAR-to-LiDAR calibration aligns points from overlapping areas of both LiDARs. The camera-LiDAR calibration involves only two parameters, making manual calibration sufficient. The process starts with initial extrinsic parameters (translation and rotation) from the CAD model. LiDAR points are colored by projecting coordinates from the LiDAR frame to the image frame, assigning corresponding pixel colors to each point.

**Table 1**
Installed sensors on the sensor platform

| Sensor | Description |
|---|---|
| Velodyne HDL-32E | Rotating LiDAR with 32 scan layers, 20Hz |
| Velodyne VLP-16 | Rotating LiDAR with 16 scan layers, 20Hz |
| Ricoh Theta V | Dual-lens 360° camera, 4k resolution, 30fps |
| Xsens MTi-10 | 3D IMU, 200Hz |



(a) Backpack (b) Common sensor rig (c) Trolley

**Figure 1:** Mobile platforms for indoor SLAM.

# 4. Digital Twin Generation

We use Google Cartographer (GC), a real-time 2D and 3D SLAM system for multiple sensor configurations and various platforms [4]. For 3D SLAM, GC is extended using 3D probabilistic grid maps. GC contains local and global SLAM algorithms, without particle filters for better performance on modest hardware. For platforms with more DOF, like the backpack, an IMU is used to project LiDAR scans into the horizontal plane, before matching each scan to a submap using non-linear optimization. The algorithm is modified to handle PCs with the additional RGB information of each point.

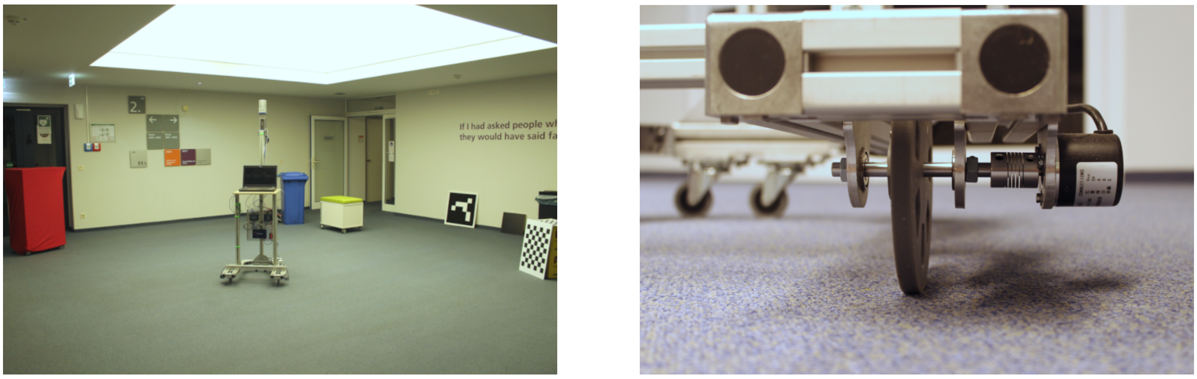
The PCs are manually divided into smaller tiles using CloudCompare [37]. This tiling process is essential for effectively partitioning the PCs into training, validation, and test splits, as well as for breaking down the labeling task into smaller steps to make it more efficient. Afterwards, a categorization of the tiles based on the corresponding rooms and corridors is performed. The name is extended with a prefix that includes the hardware platform utilized for the recording. To ensure consistency, the tiles are selected to be approximately identical across both platforms. This practice allows for performing segmentation experiments that can exclude specific areas of both platforms during training time. This tiling process results in a total of 99 tiles, where 49 tiles are from the backpack and 50 from the trolley.

To be utilized for semantic segmentation, the PCs are labeled in a manual process. For our work, 11 semantic classes are distinguished that comprise static building elements, such as *ceiling, floor, wall, beam, column, window*, and *door*, as well as items and furniture that are often found and that can be moved, such as *table, chair, sofa*, and *bookcase*. It is important to note that these classes are more fine-grained than those found in many existing indoor semantic segmentation datasets [38, 39].

Additionally, while the classes are similar to those in [40], the class *board*, from the S3DIS dataset, has been omitted for various reasons. First of all, the typical measurement error of LiDARs is very close to the thickness of the boards (typically 2-3 cm). Secondly, excluding this class accelerates the manual task of labeling. Nevertheless, the data used for our work include classes that are hard to perceive, such as closed white doors located in corridors with white walls. These door points are similar to wall points in two different ways. They share a similar geometric shape and additionally the points contain similar color information. These two factors result in complicating the annotation process.

To divide the segmentation data into training, validation and testing splits, our work uses a methodology similar to the Cityscapes benchmark [41]. For size reasons, we could not apply a more uniform distribution, e.g., as shown in the PASCAL VOC 2012 benchmark [42] (33:33:33) or sKITTI [43] (50:50 for test and train). Hence, a division of 70:15:15 is used for training, validation, and testing. Moreover, tiles are assigned to the same split if they belong to the same region within the PCs across both platforms. This is motivated by excluding certain regions during training.

For the generation of the digital twin, we use two different methods, superpoint graph (SPG) [28] and RandLA-Net [26], as they provide very good accuracy for indoor scenes [44]. SPG is a deep learning



(a) Calibration with colored objects and chessboard

(b) Wheel-encoder installation for trolley odometry

**Figure 2:** Calibration setup and trolley odometry sensor

framework for large scale PC semantic segmentation, representing PCs as graphs, interconnecting simple shapes called superpoints. The SPG classifies object parts as a whole, using superedges for contextual relationships, which are useful in supervised learning. It is much smaller than the total number of points, enabling efficient long-range interaction modeling.

The SPG is computed in an unsupervised handcrafted way. Superpoints, assumed to be semantically homogeneous, are assigned a ground truth label based on the majority label, which can cause inaccuracies. The framework includes partitioning the input PC, constructing the SPG, superpoint embedding using PointNets [25], and contextual segmentation using graph convolutions, with the latter two steps trainable in a supervised end-to-end way (Fig. 3). To improve partitioning, Landrieu and Simonovsky propose a supervised, graph-based approach for oversegmentation of PCs [45].

A second method for the digital twin generation is RandLA-Net [26], an efficient neural network architecture based on random sampling, which is chosen for its computational and memory efficiency. It applies a local feature aggregation module preserving key features and geometric details. RandLA-Net processes one million points in a single pass and is up to 200 times faster than previous methods. It shows high performance on benchmarks like S3DIS [40], Semantic3D [46], and SemanticKITTI [43].

## 5. Evaluation

To compare backpack and trolley, data from the same environments is recorded to create maps (Fig. 4). In the first area (corridor on the second floor of FOKUS, Fig. 4a and Fig. 4d), the backpack records for 419 seconds, yielding over 130 million points, while the trolley records for 363 seconds, yielding over 90 million points. In the second area (entire second floor, Fig. 4b and Fig. 4e), the backpack records for 341 seconds, producing over 103 million points, and the trolley for 414 seconds, resulting in 94 million points. In the third area (DCAITI, Fig. 4c and Fig. 4f), the backpack records for 355 seconds, yielding over 101 million points, and the trolley for 524 seconds, yielding over 155 million points.

Backpack maps show no significant qualitative loss, demonstrating the robustness of GC for the backpack. Despite the trolley's additional odometry sensor, backpack maps are comparable, indicating no significant quality gain for simpler 2D SLAM problems. However, the odometry sensor may be more important for larger areas with more loop closures. The trolley's advantage is better real-time performance due to simpler 2D scan-matching. However, for offline processing, the backpack's mobility is superior, as shown in Fig. 4c and Fig. 4f, where the trolley struggles with a small bump.

The trolley has a blind horizontal layer due to the fixed horizontally mounted LiDAR, resulting in more sparse ceiling coverage and occlusions. The coloring of the PCs is shown in Fig. 5. The backpack has better overall coverage, as seen in Fig. 5b. The visualizations demonstrate the successful fusion of LiDAR and cameras. The systematic measurement errors and different sampling rates are accounted for. For example, the car in Fig. 5c is recognizable, and the fire extinguisher is correctly colored.

PCs from both platforms show consistent coloring, due to the same sensor setup. The coloring depends on the automatic white balancing of the images, leading to variable brightness and shading in areas with changing lighting conditions. This variability is evident in the coloring of the walls. Additionally, the positional offset between camera and LiDAR can create blind spots for the camera. Since the LiDAR sensor is mounted on top of the camera, some points acquired by the LiDAR may be

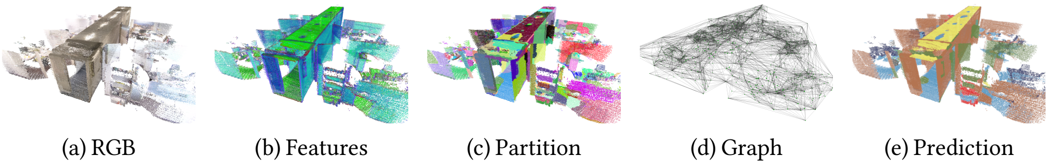
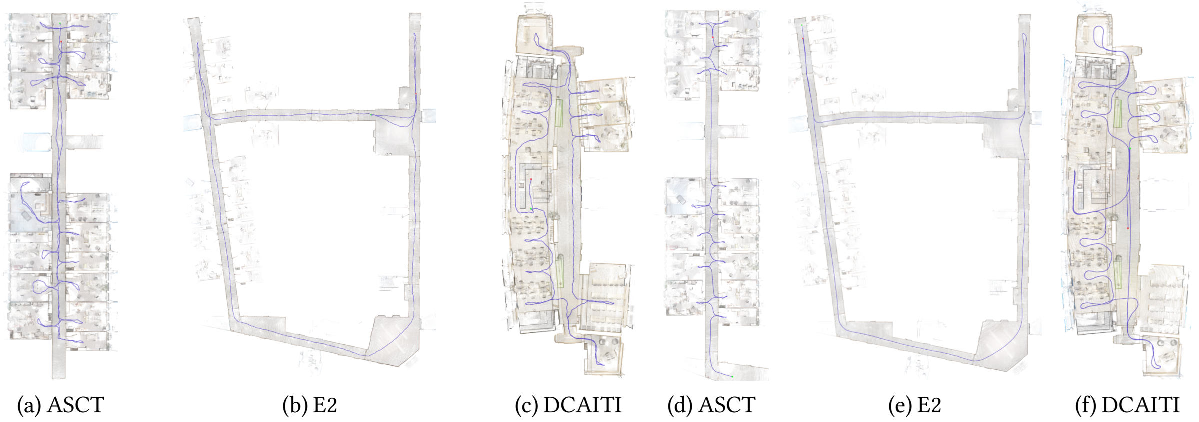
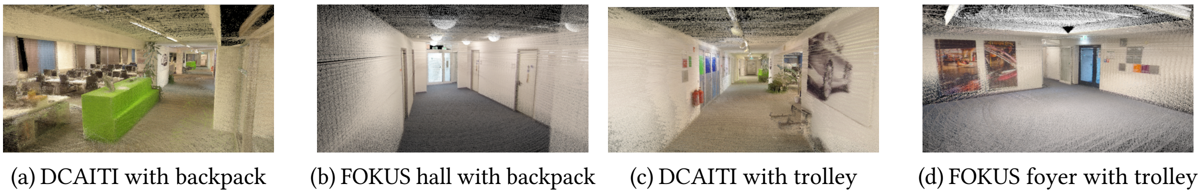

(a) RGB
(b) Features
(c) Partition
(d) Graph
(e) Prediction

**Figure 3:** Steps of the SPG framework: From the RGB input PC (a), geometric features are computed (b) to partition the PC into simple shapes (c). From this, an SPG is constructed (d). Finally, superpoints are embedded with PointNets and processed with graph convolutions to predict point-wise semantics (e).



(a) ASCT (b) E2 (c) DCAITI (d) ASCT (e) E2 (f) DCAITI

**Figure 4:** GC output for backpack (a-c) and trolley (d-f)



(a) DCAITI with backpack (b) FOKUS hall with backpack (c) DCAITI with trolley (d) FOKUS foyer with trolley

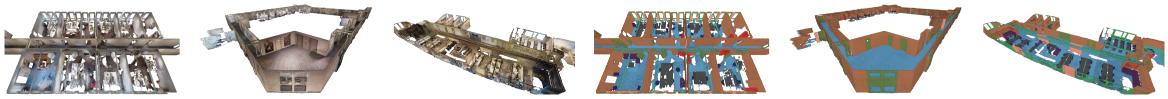
**Figure 5:** Qualitative results of the PC coloring

occluded in the images. One coloring error due to occlusions is shown in Fig. 5a, where green points on the floor and the left-hand side desk are incorrectly colored by images of the green locker.

Three training experiments for semantic segmentation are conducted for both SPG and RandLA-Net: all available training data (All), only backpack training data (BP), and only trolley training data (trolley). The overall separation yields 67.3% of the points for training, 13.4% for validation, and 19.2% for testing. Each data point has RGB and intensity values, but only RGB is used in the experiments. Intensity varies based on the laser ray's hitting angle and the object, causing variability in the same object. The void class *unlabeled* is ignored in training and testing. All semantic segmentation analyses are performed on an Intel Core i7-9700K CPU @ 3.60GHz with 64GiB of memory and an NVIDIA GeForce RTX 2080 Ti. Fig. 6 shows a qualitative assessment of ground truth annotations for all three areas.

Our evaluation compares the overall accuracy (OA), unweighted mean accuracy (mAcc), and unweighted mean intersection over union (mIoU). These metrics are evaluated for all 11 classes. The results from these quantitative measurements are illustrated in Fig. 7. In the *All* experiment, SPG achieves an overall accuracy (OA) of 68% with an mIoU of only 27%. There is a significant performance variation observed in the smaller experiments (backpack and trolley). The lowest performance occurs in the backpack experiment, resulting in an OA of 50% and an mIoU of 12%. In contrast, SPG performs best in the trolley experiment, achieving an OA of 84% and an mIoU of 43%. Notably, the SPG framework shows particularly poor performance in the backpack experiment, with certain per-class IoUs (such as for ceiling, chair, table, and bookcase) approaching zero. This indicates potential issues or errors during the training or testing phases of the framework.

When comparing both methods, RandLA-Net consistently outperforms SPG across all experiments. Even in the most successful SPG experiment (trolley), RandLA-Net achieves an overall accuracy (OA) that is 6% higher, with an even larger difference for mIoU. Additionally, RandLA-Net demonstrates significantly faster computation times compared to the SPG framework, aligning with the results presented in [26]. However, for the practical application of automatically generating semantically segmented PCs, SPG performs reasonably fast on the used hardware.



(a) RGB col. ASCT (b) RGB col. E2 (c) RGB col. DCAITI (d) GT ASCT (e) GT E2 (f) GT DCAITI

**Figure 6:** RGB colored (col.) point clouds and the annotated ground truth (GT) clouds

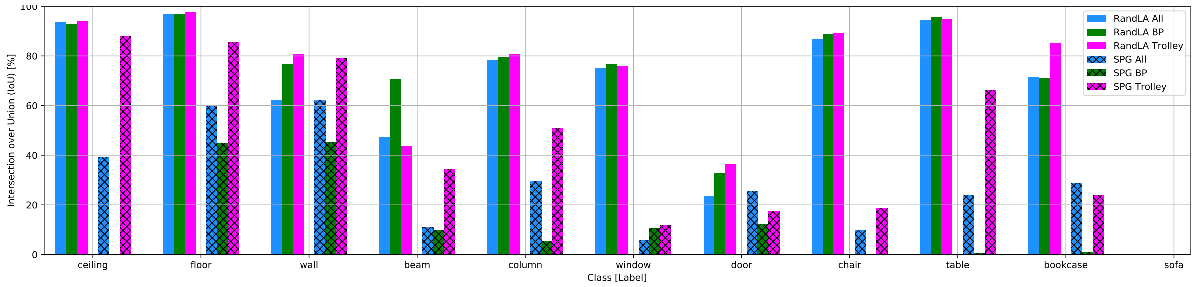


**Figure 7:** Experimental test results of SPG and RandLA-Net. Intersection over union metric is shown per class.

## 6. Conclusion

This paper presents two state-of-the-art deep neural networks for semantic segmentation to create digital twins for different indoor environments. A novel modular sensor platform with two LiDARs, an industrial-grade IMU, and a dual-lens fish-eye camera, produces high-quality PCs, enabling LiDAR measurements to be colored from camera images. Two experimental platforms are designed: a lightweight backpack for non-planar environments and a trolley for planar movements. Six PCs covering three areas are produced, totaling over 41 minutes of raw sensor data and 673 million points. Analysis show that the backpack is superior for this task due to its mobility. Two deep neural network architectures are trained and evaluated. The SPG framework reveals issues with backpack PCs but provides valuable insights. RandLA-Net outperforms the SPG framework, achieving 89.7% overall Acc and an mIoU of 70.74% on trolley data, demonstrating the effectiveness of deep neural networks for digital twin creation.

A primary issue in supervised machine learning is the lack of training data. Future work could extend the training dataset using the developed sensor platform. Current settings from Google Cartographer can be applied, and label classes could be extended to improve coverage. Analyzing unlabeled points and creating new classes could enhance data coverage. Currently, the indoor digital twin exists as 3D environment that can be accessed with a browser. A next step could be the deduction of a 2D map as it is used for routing in [1] by 3D-to-2D projection.

## Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o mini in order to: Grammar and spelling check, Paraphrase and reword. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] J. Wortmann, B. Schäufele, K. Klipp, I. Radusch, K. Blaß, T. Jung, Enhanced accessibility for mobile indoor navigation, in: 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2024.

[2] K. Massow, B. Kwella, N. Pfeifer, F. Häusler, J. Pontow, I. Radusch, J. Hipp, F. Dölitzscher, M. Haueis, Deriving HD maps for highly automated driving from vehicular probe data, in: IEEE 19th International Conference on Intelligent Transportation Systems, IEEE, 2016.

[3] B. Henke, J. N. Hark, D. Becker, O. Sawade, I. Radusch, Map Switching Monte Carlo LiDAR Localization for Automated Driving in Parking Garages, in: Intelligent Vehicles Symposium, IEEE, 2019.

[4] W. Hess, D. Kohler, H. Rapp, D. Andor, Real-time loop closure in 2D LiDAR SLAM, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 1271–1278.

[5] R. Otero, S. Lagüela, I. Garrido, P. Arias, Mobile indoor mapping technologies: A review, Automation in Construction 120 (2020).

[6] M. Bosse, R. Zlot, P. Flick, Zebedee: Design of a Spring-Mounted 3-D Range Sensor with Application to Mobile Mapping, IEEE Transactions on Robotics 28 (2012) 1104–1119.

[7] A. Nüchter, M. Bleier, J. Schauer, P. Janotta, Improving Google's Cartographer 3D mapping by continuous-time slam, The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2017) 543.

[8] D. Chetverikov, D. Svirko, D. Stepanov, P. Krsek, The trimmed iterative closest point algorithm, in: 2002 International Conference on Pattern Recognition, volume 3, IEEE, 2002, pp. 545–548.

[9] J. Zhang, S. Singh, LOAM: Lidar Odometry and Mapping in Real-time, in: Robotics: Science and Systems, volume 2, 2014, p. 9.

[10] J. Zhang, S. Singh, Visual-lidar odometry and mapping: Low-drift, robust, and fast, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 2174–2181.

[11] G. D. Tipaldi, M. Braun, K. O. Arras, FLIRT: Interest Regions for 2D Range Data with Applications to Robot Navigation, in: Experimental Robotics, Springer, 2014, pp. 695–710.

[12] M. Himstedt, J. Frost, S. Hellbach, H.-J. Böhme, E. Maehle, Large scale place recognition in 2D LiDAR scans using geometrical landmark relations, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 5030–5035.

[13] K. Granström, T. B. Schön, J. I. Nieto, F. T. Ramos, Learning to close loops from range data, The International Journal of Robotics Research 30 (2011) 1728–1754.

[14] G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard, A tutorial on graph-based SLAM, IEEE Intelligent Transportation Systems Magazine 2 (2010) 31–43.

[15] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, G$^2$o: A general framework for graph optimization, in: IEEE International Conference on Robotics and Automation, 2011.

[16] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, C. Stachniss, SuMa++: Efficient LiDAR-based Semantic SLAM, in: International Conf. on Intelligent Robots and Systems, IEEE, 2019.

[17] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. J. Leonard, Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, IEEE Transactions on Robotics 32 (2016) 1309–1332.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: European conference on computer vision, 2018.

[19] J. Yuan, Z. Deng, S. Wang, Z. Luo, Multi receptive field network for semantic segmentation, in: IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2020, pp. 1883–1892.

[20] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, Q. V. Le, Rethinking pre-training and self-training, Advances in Neural Information Processing Systems 2020-December (2020).

[21] E. Grilli, F. Menna, F. Remondino, A review of point clouds segmentation and classification algorithms, The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 42 (2017) 339.

[22] H. Lu, H. Shi, Deep Learning for 3D Point Cloud Understanding: A Survey, arXiv preprint arXiv:2009.08920 (2021).

[23] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, M. Felsberg, Deep projective 3D semantic segmentation, in: International Conference on Computer Analysis of Images and Patterns, Springer, 2017, pp. 95–107.

[24] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, S. Savarese, Segcloud: Semantic segmentation of 3d point clouds, in: International Conference on 3D Vision (3DV), IEEE, 2017, pp. 537–547.

[25] C. R. Qi, H. Su, K. Mo, L. J. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[26] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, RandLA-Net: Efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11108–11117.

[27] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, L. J. Guibas, KPConv: Flexible and Deformable Convolution for Point Clouds, in: International Conference on Computer Vision, IEEE, 2019.

[28] L. Landrieu, M. Simonovsky, Large-scale point cloud semantic segmentation with superpoint graphs, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4558–4567.

[29] Z. Xu, X. Huang, B. Yuan, Y. Wang, Q. Zhang, W. Li, X. Gao, Retrieval-and-alignment based large-scale indoor point cloud semantic segmentation, Science China Information Sciences 67 (2024).

[30] C. Choy, J. Park, V. Koltun, Fully convolutional geometric features, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 8958–8966.

[31] D. Rozenberszki, O. Litany, A. Dai, UnScene3D: Unsupervised 3D Instance Segmentation for Indoor Scenes, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024.

[32] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, B. Leibe, Mask3D: Mask Transformer for 3D Semantic Instance Segmentation, in: IEEE International Conference on Robotics and Automation, 2023.

[33] Y.-Q. Yang, Y.-X. Guo, J.-Y. Xiong, Y. Liu, H. Pan, P.-S. Wang, X. Tong, B. Guo, Swin3D: A pretrained transformer backbone for 3D indoor scene understanding, Computational Visual Media 11 (2025).

[34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[35] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, Z. Zhou, Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling, in: Computer Vision – ECCV 2020, Springer, 2020.

[36] A. Geiger, F. Moosmann, Ömer Car, B. Schuster, Automatic camera and range sensor calibration using a single shot, in: IEEE International Conference on Robotics and Automation, 2012.

[37] D. Girardeau-Montaut, CloudCompare, https://www.danielgm.net/cc, 2016. Retrieved from CloudCompare.

[38] J. Xiao, A. Owens, A. Torralba, SUN3D: A Database of Big Spaces Reconstructed using SfM and Object Labels, in: Proceedings of the IEEE International Conference on Computer Vision, 2013.

[39] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), IEEE, 2011.

[40] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D Semantic Parsing of Large-Scale Indoor Spaces, in: IEEE Conf. on Computer Vision and Pattern Recognition, 2016.

[41] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[42] M. Everingham, S. A. Eslami, L. V. Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International Journal of Computer Vision 111 (2015) 98–136.

[43] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, J. Gall, SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences, in: IEEE International Conference on Computer Vision, 2019, pp. 9297–9307.

[44] Y. Sun, X. Zhang, Y. Miao, A review of point cloud segmentation for understanding 3D indoor scenes, Visual Intelligence 2 (2024) 14.

[45] L. Landrieu, M. Boussaha, Point cloud oversegmentation with graph-structured deep metric learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[46] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, M. Pollefeys, Semantic3D.net: A new Large-scale Point Cloud Classification Benchmark, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, volume IV-1-W1, 2017.