Comparison of Point Cloud and Surface Based Mapping for Autonomous Vehicles

Christoph Brückner[a] and Lucila Patino-Studencki[b]
Faculty Mechanical and Automotive Engineering, Coburg University of Applied Sciences and Art, Güterstraße 8, Kronach, Germany
{*christoph.brueckener, lucila.patino-studencki*}*@hs-coburg.de*

Keywords: Mapping, SLAM, Localization, Autonomous Driving, Robotics.

Abstract: Mapping and localization are essential processes in robotics and autonomous systems, providing precise environmental representation and real-time positioning. Unlike Simultaneous Localization and Mapping (SLAM), which combines these tasks simultaneously, mapping and localization are often decoupled in applications that require higher accuracy and efficiency from the outset, like autonomous vehicles. This study summarizes the main families of map representations used in SLAM and investigates the applicability for standalone mapping and localization tasks. Point cloud and surfaced based Mapping Methods, namely KISS-ICP and PUMA are explored and evaluated numerically using the KITTI database. Key performance metrics accuracy, registration time during localization, and map size are analyzed to compare their effectiveness. The results provide insights into the strengths and limitations of SLAM-based techniques when applied to decoupled processes.

1 INTRODUCTION

Autonomous vehicles (AV) are robots on the street, which means that they are Machines (programmable by a computer) that can move independently and perform complex actions similar to a living creature interacting with its environment. In the case of AVs their environment is defined by the road network and other traffic participants, such as other vehicles, pedestrians, and road infrastructure. In this environment, vehicles must operate reliably. To achieve this, the vehicle must accurately determine its position (localization) and understand the constraints imposed by its environment, that is, the mapping of static structures such as road boundaries, buildings, and trees.

Both localization and mapping are closely related and are mainly addressed in the robotics literature under the theme of simultaneous localization and mapping (SLAM). Therefore, this paper addresses various mapping methods used in robotic applications and compares them based on how they represent the environment, as well as the localization accuracy they provide. The analysis focuses on "traditional" methods and excludes artificial intelligence (AI) approaches, since their implementation is highly specific and their capabilities require a significant amount of data. However, the presented map representations are equally valid for AI approaches.

In the literature, some comparisons of Simultaneous Localization And Mapping (SLAM) approaches exist. Some focus on a single family of representations like Iterative Closest Point (ICP), as in Pomerleau et al. (Pomerleau et al., 2015), or land-mark-based maps, as in Dai et al. (Dai et al., 2023) and Debeunne and Vivet (Debeunne and Vivet, 2020). Other reviews compare multiple map representations (e.g. (Schreier et al., 2015), an (Bao et al., 2023)), but these are typically qualitative rather than quantitative. As a result, these studies are limited to analyze the characteristics of the representations without performing any numerical evaluations. In this paper, a numerical evaluation of various SLAM methods adapted for autonomous vehicles using the KITTI dataset (Geiger et al., 2012) is presented.

The rest of this paper is organized as follows. section 2 reviews the most common representations and mapping methods used in robotics, while section 3 emphasizes the algorithmic differences and constraints of the mapping for autonomous driving. section 4 compares two of the methods numerically with respect to positioning accuracy and memory requirements. The last section presents the main conclusions of the analysis.

[a] https://orcid.org/0009-0006-3378-0033
[b] https://orcid.org/0009-0002-5349-422X

Brückner, C. and Patino-Studencki, L.
Comparison of Point Cloud and Surface Based Mapping for Autonomous Vehicles.
DOI: 10.5220/0013473000003941

In *Proceedings of the 11th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2025)*, pages 199-209
ISBN: 978-989-758-745-0; ISSN: 2184-495X

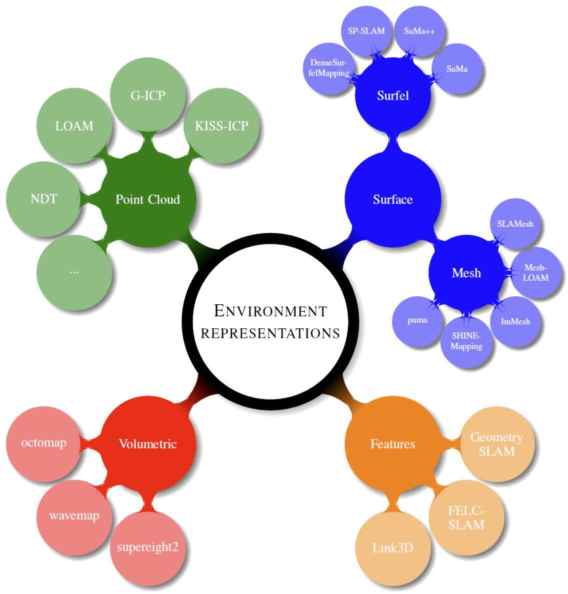

Figure 1: Overview of environment representation techniques and mapping Implementations.

2 OVERVIEW OF MAPPING METHODS

Mapping real-world environments is essential for autonomous vehicle (AV) systems, with different methods offering unique strengths and limitations to represent the environment. Four main types of representation are found in the literature, namely: Point Clouds, Surfaces, Volumetric and Features maps.

A summary of these different representation techniques and some of their implementations in the field of robotics and autonomous driving is shown in Figure 1. In the following the mentioned methods and approaches to create the environment representation (map) are described.

2.1 Volumetric Based Mapping Approaches

Volumetric-based mapping methods are common in robotics, offering robust 3D environment representation by dividing space into volumetric elements called voxels. These discretize space into 3D grids, excel in volumetric occupancy mapping and are well suited for collision avoidance. Their major drawback is high memory usage at finer resolutions. Methods like OctoMap (Hornung et al., 2013; Wurm et al., 2010), WaveMap (Reijgwart et al., 2023), and the method of Vespa et al. (Vespa et al., 2018; Funk et al., 2021; Vespa et al., 2019) known as supereight2. They differ in how they process and store information, presenting distinct challenges and trade-offs in memory usage, accuracy, and localization capabilities. OctoMap is a widely-used framework that employs an octree structure to hierarchically subdivide 3D space into cubic volumes (voxels) until a given minimum voxel size. It efficiently stores data of empty or occupied regions while reducing memory usage. However, its reliance on minimal fixed-resolution voxels can limit detail in high-resolution areas, affecting map accuracy. WaveMap introduces wavelet-based encoding (Yguel et al., 2006) for volumetric representation, enabling multiscale resolution within the same map. This adaptive approach preserves detail in critical regions while minimizing memory usage, making it more efficient than OctoMap in handling large, complex environments. Furthermore, the wavelet encoding and reconstruction processes are computationally less intensive than the reconstruction of a high resolution Octomap. WaveMap enhances localization by capturing finer details and offering better adaptability to environmental changes. Supereight2 uses voxel block to dynamically allocate memory only to regions of interest, means the region from empty to occupied space. It achieving efficient memory management even at high resolutions. It supports dense mapping and real-time updates, making it suitable for dynamic urban environments. Supereight2's dense representation supports precise localization, especially in feature-rich settings. Volumetric methods face challenges in balancing resolution, memory efficiency, and real-time performance.

2.2 Feature Based Mapping Approaches

Feature-based mapping methods offer an alternative approach in autonomous driving, known for their ability to create compact and computationally efficient maps by extracting key features (e.g., edges, corners) from the environment offering computational efficiency and compact map storage. However, these methods may lose environmental details, making them less effective in highly dynamic or feature-poor environments. Methods like Link3D (Cui et al., 2024; Cui et al., 2023), FELC-SLAM (Gao et al., 2024), and GeometrySLAM (Xu et al., 2023) each offer unique approaches to balance accuracy, memory usage, and localization capabilities. Link3D emphasizes lightweight mapping by connecting geometric features, such as edges and planes, into a sparse network. This reduces memory requirements significantly and provides efficient localization. However, the sparse representation may lose critical details in complex environments, limiting its adaptability in dynamic or unstructured scenarios. FELC-SLAM builds on feature extraction by incorporating edge and line constraints into the mapping process. It enhances localization accuracy, particularly in urban environments where linear features are prevalent. While FELC-SLAM achieves a good balance between map detail and memory usage, it struggles with real-time performance in highly dynamic areas due to its reliance on feature stability. GeometrySLAM focuses on extracting and tracking geometric primitives, such as corners and planes, for efficient mapping and robust localization. Its compact representation minimizes memory usage, but challenges arise in feature-poor areas where insufficient primitives can reduce map quality and localization reliability. Feature-based methods are memory-efficient and provide strong localization capabilities, especially in structured environments. However, their reliance on sparse features can limit performance in dynamic or unstructured settings, requiring careful method selection for autonomous driving applications.

2.3 Surface Based Mapping Approaches

Surface-based mapping methods are another solution for environment representation in autonomous driving, representing environments as continuous surfaces using mesh or surfel-based approaches. These methods offer detailed and structured maps but vary in memory usage, computational demands, and localization capabilities. Mesh-based approaches create triangular meshes to represent surfaces and aresuitable for path planning, though generating and maintaining them in real-time can be computationally demanding. Poisson Surface Reconstruction for LiDAR Odometry and Mapping known as PUMA (Vizzo et al., 2021) and SHINE-Mapping (Zhong et al., 2023) focuses on real-time mesh generation with high accuracy but faces challenges in computational efficiency and scalability in large environments. ImMesh (Lin et al., 2023) and Mesh-LOAM (Zhu et al., 2024) improve upon this by integrating efficient incremental updates and feature-based optimizations, enabling better memory management and real-time performance. SLAMesh (Ruan et al., 2023) combines mesh creation with SLAM techniques, offering strong localization through loop closure and global optimization but at the cost of higher memory usage for maintaining global meshes.

Surfel-based approaches use discrete surface elements (surfels), which stores additional attributes like normal vectors, to represent environment surfaces. SuMa (Behley and Stachniss, 2018) and its enhanced semantic version, SuMa++ (Chen et al., 2019), use LiDAR data to build dense surfel maps, achieving precise localization with efficient use of memory. However, maintaining consistency over large areas remains a challenge. SP-SLAM incorporates color features into surfel maps, enhancing localization in dynamic environments but requiring more computational resources. DenseSurfelMapping (Wang et al., 2019) focuses on high-resolution surfel representation, offering detailed maps and robust localization but at the expense of increased memory usage. Both mesh and surfel methods excel in creating accurate surface maps, with mesh approaches favoring structured representations for global consistency and surfel methods emphasizing real-time performance. However, managing memory and computational demands is a common challenge, particularly for large-scale or dynamic environments.

2.4 Point Cloud Based Mapping Approaches

Point cloud representations, composed of discrete 3D points from LiDAR or cameras, provide high-resolution geometric data, making them ideal for detailed environmental representation. However, they require significant storage and processing resources on large scales. Key methods to create a point cloud map, such as Iterative Closest Point (ICP), and its newer, enhanced variants like Generalized Iterative Closest Point ICP (G-ICP) (Segal et al., 2009; Moon et al., 2024), Keep It Simple Straightforward Iterative Closest Point (KISS-ICP) (Vizzo et al., 2023), LOAM (Zhang and Singh, 2014), and Normal Distributions Transform (NDT) (Biber and Strasser, 2003; Saarinen et al., 2013) differ slightly in the map representation used, memory usage, and suitability for localization, especially when incorporating downsampling or feature detection for memory efficiency. G-ICP enhances alignment precision by combining point-to-point and point-to-plane metrics, and including covariances of each point excelling in structured environments. However, its computational and memory demands are high due to the search for the nearest neighbors of every point before computing covariances (Young et al., 2021). Downsampling can alleviate memory usage but may reduce accuracy in feature-rich areas. G-ICP offers reliable localization because of the additional information taken into account. KISS-ICP focuses on incremental point cloud alignment using simple nearest-neighbor matching, making it computationally efficient and memory-light. While suitable for real-time tasks, it struggles in complex or noisy environments. Downsampling helps reduce processing time but risks losing critical details, affecting localization. LOAM separates odometry and mapping, extracting features such as edges and planes for accurate alignment. The accuracy in the mapping algorithm is ensured through feature matching, which also facilitates fast computation in the odometry algorithm (Zhang and Singh, 2014). It balances accuracy and real-time performance, making it ideal for urban scenarios, but feature extraction and global map maintenance are memory-intensive. Downsampling helps manage data size, but can weaken feature extraction, affecting map detail and localization. NDT divides the space in a grid and represents the collected point cloud as Gaussian distribution saving only their parameters, matching distributions instead of individual points. It is computationally efficient in structured environments and resilient to noise but has high memory demands for storing grids and struggles in sparse or dynamic areas (Pang et al., 2018). Downsampling reduces memory load, but may compromise grid quality and alignment accuracy. Downsampling is essential for managing memory and computational costs across these methods, but it trades off map resolution and alignment accuracy. Proper trade-offs between memory usage and localization robustness are key for autonomous driving applications. These aspects are analyzed later in section 4.

3 MAPPING AND LOCALIZATION FOR AUTONOMOUS VEHICLES

In robotics Simultaneous Localization and Mapping (SLAM) is a widely used approach , enabling systems to navigate unknown environments by simultaneously building a map and determining their position within it. SLAM is particularly effective for exploration tasks, as it dynamically updates maps in real time to reflect environmental changes. However, SLAM comes with limitations, including reduced accuracy during the initial convergence phase and significant computational demands to maintain real-time performance. These challenges make SLAM less suitable for applications like autonomous driving, where precise positioning and high-resolution environmental representation are required from the outset.

In contrast, mapping and localization are decoupled processes that address specific challenges independently. The map shall be created in a process previous to the navigation phase, when the localization take place. This separation allows each process to be optimized for its unique requirements. Mapping focuses on creating a static representation of the environment, while localization determines the system's position in real time relative to the precomputed map. Moreover, since the map is not created by the same vehicle using the same sensors long stability of the map representation needs to be ensured.

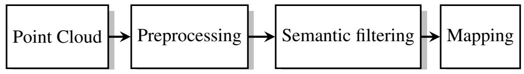

Figure 2: Process of mapping.

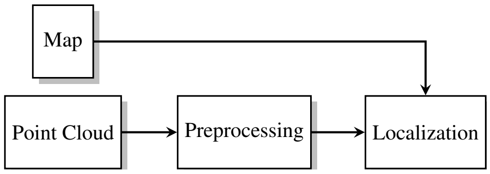

Figure 3: Process of localization.

The mapping, as shown in Fig. 2 is typically performed offline, allowing for the use of computationally intensive techniques to achieve high-resolution results. Point cloud data from LIDAR or cameras serves as the input, undergoing preprocessing steps like deskewing to correct distortions in sensor measurements. Additionally, semantic filtering is applied to remove dynamic elements, such as vehicles, and retain only static objects like buildings and roads. This ensures that the generated maps are both accurate and reliable for navigation or simulation purposes. However, due to its offline nature, mapping lacks adaptability and cannot dynamically update to reflect changes in the environment.

The localization process, as shown in Fig. 3, on the other hand, operates in real time and relies on the precomputed map to determine the system's precise position. Point cloud data is preprocessed to ensure accuracy, and semantic filtering may also be applied to refine the input further. Localization's real-time nature is crucial for dynamic tasks in autonomous driving, where the ability to make decisions quickly and interact with other traffic participants is essential.

In applications like autonomous vehicles, mapping and localization are performed at distinct time points. Mapping occurs during the predeployment phase, often utilizing pre-recorded, high-accuracy GPS data (e.g., RTK) to create detailed maps. Localization is then carried out during operation, relying on these precomputed maps to provide precise and efficient positioning. This separation eliminates the need for a convergence phase, ensuring that vehicles know their position accurately from the outset.

4 EVALUATION OF SLAM METHODS FOR MAPPING AND LOCALIZATION APPROACHES

After summarizing the characteristics of map representations and their main approaches, their numerical evaluation is critical for understanding their performance and suitability for autonomous driving. In this study, we perform experiments two of the most significative algorithms presented in section 2, namely: the KISS-ICP, and PUMA SLAM, analyzing their capabilities on a subset of KITTI benchmark sequences (Geiger et al., 2012). These algorithms represent a two of the main SLAM approaches, each with unique characteristics in terms of algorithm design and output.

Our evaluation focuses on three key metrics: pose accuracy, registration time during localization, and relative map size. Accuracy assesses the precision of localization and the quality of the generated maps. Registration time measures the computational efficiency of each method, highlighting their suitability for real-time applications. Map size, in the context of map accuracy, evaluates how efficiently the methods encode spatial information, balancing storage requirements with fidelity. By systematically comparing these SLAM packages, we aim to identify their strengths, limitations, and trade-offs, providing valuable insights into their application in mapping and localization tasks for robotics and autonomous systems.

4.1 Testing Environment

For evaluating the algorithms a testing framework based on the well-known KITTI dataset of odometry (Geiger et al., 2012) was chosen. This dataset provides high-quality point cloud data with preprocessing steps (mentioned in chapter 3) already included, ensuring a reliable foundation for experimentation. The processes of mapping and localization are divided in similar way as done in autonomous vehicles instead of using a SLAM approach. For mapping, we use the Semantic KITTI (Behley et al., 2019; Behley et al., 2021) dataset, an extension of the KITTI dataset enriched with semantic annotations. This enables us to apply semantic filtering during the mapping process, removing dynamic elements such as vehicles while retaining only static objects to create more accurate and consistent maps. To generate the map, we leverage the ground truth poses provided by the KITTI dataset, utilizing an adapted version of the algorithm tailored specifically for this purpose. From the dataset, we chose 4 random data sequences for the analysis to include some diversity while keeping the computation time within limits.

For localization, we use the original KITTI dataset without additional semantic filtering. This simplifies the localization pipeline by retaining dynamic objects in the data while focusing on real-time positioning relative to the precomputed maps. Using these complementary datasets, we ensure a robust test environment to assess the performance of the approaches in addressing the distinct requirements of mapping and localization.

The experiments are conducted on a system running Ubuntu 22.04 within Windows Subsystem for Linux (WSL) on a Windows 11 host. The hardware setup features an AMD Ryzen 7 5700G processor and 32 GB of RAM, providing sufficient computational resources to handle large point cloud datasets. In the next subsections the results for each of the algorithms are presented.

4.2 Point Cloud Based Mapping

In this evaluation, we assess the performance of Point Cloud Map by using KISS-ICP (Vizzo et al., 2023) on selected KITTI benchmark sequences to understand its effectiveness in mapping and localization tasks. The evaluation is conducted using the default configuration by adapting the maximum number of threads to 8 and increase the maximum number of iterations to 1,000 to improve convergence accuracy during the scan alignment process, particularly in complex environments. The voxel size is adjusted in accordance with the map voxelization, ensuring consistency. An impression of the wide range of voxel sizes and the resulting point cloud maps is shown in Fig. 4, with the zoomed scene below. The larger the voxel, the fewer points are used to represent the map, resulting in a lighter color.

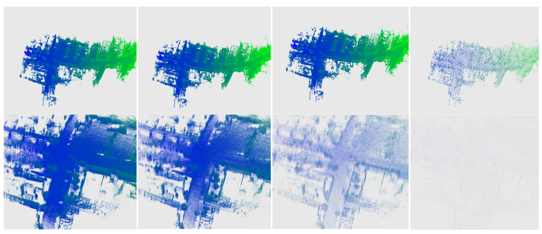

Figure 4: Point cloud map of sequence 03 at different voxel sizes; from left to right [3 cm, 10 cm, 25 cm, 100 cm].

Figure 5 presents the localization accuracy results, including absolute and relative translational errors (difference between the estimated and actual movement of the vehicle between two points in time), as well as absolute and relative rotational errors (angular difference). These metrics are computed using

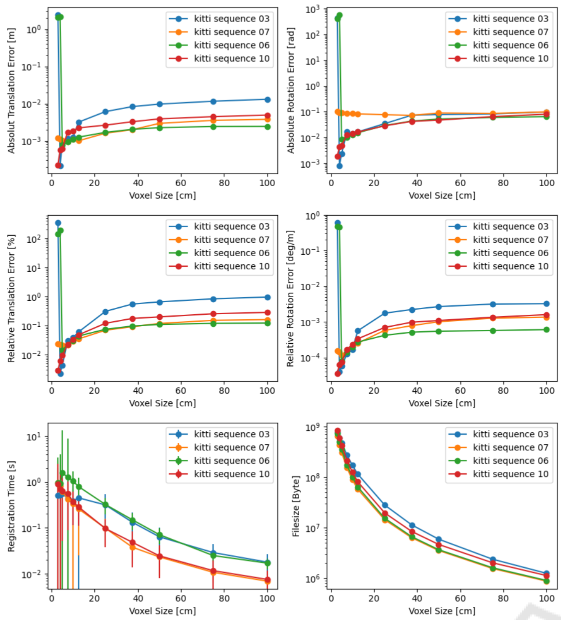

Figure 5: Results of KISS-ICP.

the KISS-ICP evaluation framework, which closely aligns with the metrics employed in the KITTI odometry benchmark. In the lower part of the figure the registration time and the file size of the map are depicted. Registration time is the time that the algorithm needs for calculate the correspondences (registration) between the sensor data and the map.

In general, it can be said that the accuracy improves as the voxel size decreases, at least up to a sequence-specific threshold where the process becomes unstable. Beyond this point, finer voxel grids result in a significant decline in accuracy, producing poor outcomes. This behavior is also evident in the figures presented in the appendix and the trajectory plots of the sequences, where unstable attempts exhibit oscillations and diverge completely from the intended path. The threshold at which instability occurs appears to be dependent on the specific map or sequence. Some sequences remain stable with voxel sizes as small as 3 cm—the lower limit of this investigation—while others shows instability at voxel size of 3 and 4 cm.

Overall, the results demonstrate that good accuracy can still be achieved even at lower resolutions of the point cloud map, e.g. a translation error below 1 cm is achievable for voxel sizes of 50 cm. As expected, computation times and map sizes increase as the voxel size decreases. Between the lowest resolution tested (100 cm voxel size) and the highest resolution (3 cm voxel size), processing time increases by approximately 100 times. Similarly, map size shows a rapid exponential growth with increasing resolution, underscoring the trade-off between finer detail for higher accuracy and computational and storage demands.

4.3 Surface Based Mapping

The second approach we explore is the PUMA algorithm (Vizzo et al., 2021) for mapping and localization tasks. By leveraging the default properties of the PUMA package, we adapted a SLAM (Simultaneous Localization and Mapping) approach to the previously proposed mapping and localization framework.

A notable aspect of this adaptation involves controlling the map resolution. This is accomplished by adjusting the depth parameter of the PUMA algorithm, which controls the granularity of the mesh and, consequently, the resolution of the map, as can seen in Fig. 6. This allows for the representation of the environment at varying levels of detail, ranging from coarser to finer meshes.

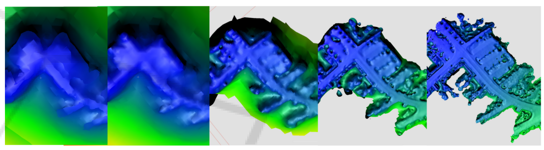

Figure 6: Mesh Map of sequence 03 at different depth values; from left to right [6, 7, 8, 9, 10].

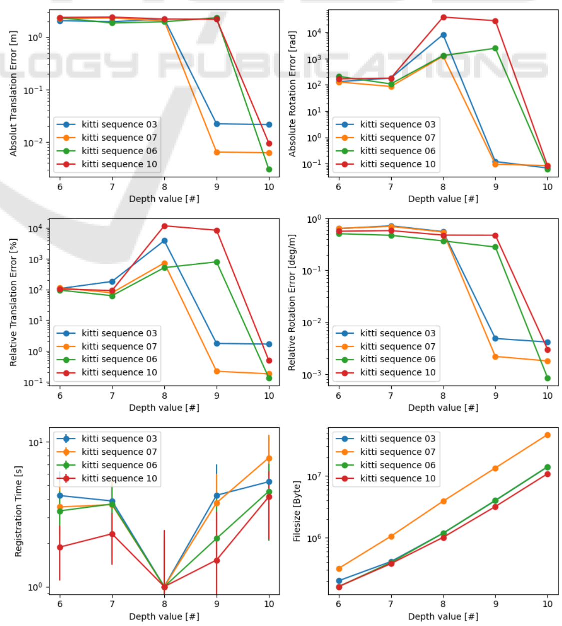

Figure 7: Results of PUMA.

Figure 7 highlights the performance characteristics of the PUMA algorithm under varying depth values. Notably, all sequences are successfully resolved only when the depth value reaches 10, although some sequences are also solvable with a depth value of 9, as further illustrated in the trajectory plots provided in the appendix. The computation time varies significantly, ranging between 1 and 10 seconds. At a depth value of 8, the computation time unexpectedly drops to around 1 second for all sequences. However, this is accompanied by an increase in both the relative translation error and the absolute rotation error. Combined with the trajectory plots in the appendix, this suggests that the localization process fails from the very first iteration. The underlying cause of this consistent behavior across all sequences at the same depth value remains unclear and warrants further investigation.

The file size increases exponentially with the depth value, reaching 10 MB to 50 MB at a depth value of 10, depending on the sequence length. In terms of accuracy, the overall absolute translation error achieves a minimum value below 3 cm the depth value of 10 for all sequences. Further parameter optimization may help reduce these errors, potentially enhancing both accuracy and efficiency.

4.4 Comparison of the Approaches

When comparing the two methods based on the chosen metrics—accuracy, registration time, and map size—the point cloud representation with KISS-ICP clearly stands out. It achieves the highest accuracy, reaching theoretically sub-millimeter levels, and significantly smaller file sizes relative to the achieved accuracy. For instance, achieving centimeter-level accuracy with PUMA requires a file size of 10 to 50 MB, depending on the sequence, whereas a similar level of accuracy with the point-cloud representation can be achieved with file sizes below 7 MB. Registration time defines the real-time position availability too. KISS-ICP demonstrates the fastest registration times, approximately 10 milliseconds for large voxel sizes, highlighting its potential for real-world applications. Additionally, the PUMA localization seems to be more unstable compared to the point cloud localization, the frequent divergence of some of the sequences depending on the depth value demonstrates the susceptibility of localization stability to the selected parameters.

Despite these differences, mesh maps offer an advantage in their inherent topography information, which facilitates path planning by enabling collision-free navigation. In contrast, unstructured point cloud maps lack this built-in topographical data, making collision-free path planning more challenging in respect of computational effort.

5 CONCLUSIONS

The analysis of the characteristics of different environment representations: point clouds, surface-based models such as meshes and surfels, volumetric maps, and feature-based maps bring us to the conclusion that although SLAM approaches are suitable for robotics applications they require some adaptions for dividing localization and mapping as required in autonomous vehicles.

Our comparative numerical analysis of adapted SLAM approaches demonstrated that point cloud-based mapping using KISS-ICP offers significant advantages over mesh-based methods. Specifically, the point cloud approach excelled in terms of accuracy, computational efficiency, stability, and reduced map size. These findings highlight the suitability of point clouds for scenarios that demand high precision and real-time performance, such as autonomous driving.

Looking to the future, there is considerable potential for further advancements in this field. Expanding the analysis to include other representation methods, such as feature-based and volumetric maps, could provide a more comprehensive understanding of the algorithms and allow a efficient evaluation of their use-cases. Extending the evaluation to include other datasets and exploring alternative alignment (registration) algorithms will further enhance the understanding of the trade-offs between these approaches and make possible to build more robust algorithms for autonomous driving.

ACKNOWLEDGEMENTS

The project 'DEKOR-X' is part of the 'Flagship Initiative for Autonomous and Connected Driving' of the German Association of the Automotive Industry (VDA) and is being funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK). We sincerely thank all project partners and contributors for their valuable support and collaboration throughout this research.

REFERENCES

Bao, Z., Hossain, S., Lang, H., and Lin, X. (2023). A review of high-definition map creation methods for autonomous driving. *Engineering Applications of Artificial Intelligence*, 122:106125.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Gall, J., and Stachniss, C. (2021). Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset.

The International Journal of Robotics Research, 40(8-9):959–967.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., and Gall, J. (2019). SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. Version Number: 3.

Behley, J. and Stachniss, C. (2018). Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation.

Biber, P. and Strasser, W. (2003). The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2743–2748, Las Vegas, Nevada, USA. IEEE.

Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., and Stachniss, C. (2019). SuMa++: Efficient LiDAR-based Semantic SLAM. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537, Macau, China. IEEE.

Cui, Y., Chen, X., Zhang, Y., Dong, J., Wu, Q., and Zhu, F. (2023). BoW3D: Bag of Words for Real-Time Loop Closing in 3D LiDAR SLAM. *IEEE Robotics and Automation Letters*, 8(5):2828–2835.

Cui, Y., Zhang, Y., Dong, J., Sun, H., Chen, X., and Zhu, F. (2024). LinK3D: Linear Keypoints Representation for 3D LiDAR Point Cloud. *IEEE Robotics and Automation Letters*, 9(3):2128–2135.

Dai, Y., Wu, J., and Wang, D. (2023). A review of common techniques for visual simultaneous localization and mapping. *Journal of Robotics*, 2023(1):8872822.

Debeunne, C. and Vivet, D. (2020). A review of visual-lidar fusion based simultaneous localization and mapping. *Sensors*, 20(7):2068.

Funk, N., Tarrio, J., Papatheodorou, S., Popović, M., Alcantarilla, P. F., and Leutenegger, S. (2021). Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning. *IEEE Robotics and Automation Letters*, 6(2):3553–3560.

Gao, R., Li, Y., Li, B., and Li, G. (2024). FELC-SLAM: feature extraction and loop closure optimized lidar SLAM system. *Measurement Science and Technology*, 35(11):115112.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, Providence, RI. IEEE.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206.

Lin, J., Yuan, C., Cai, Y., Li, H., Ren, Y., Zou, Y., Hong, X., and Zhang, F. (2023). ImMesh: An Immediate LiDAR Localization and Meshing Framework. *IEEE Transactions on Robotics*, 39(6):4312–4331.

Moon, Y., Hong, J., Park, C., and Han, S. (2024). Faster GICP with Voxel-Based Computation: Dynamic voxel size refinement for Enhanced Accuracy. In *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, pages 493–498, Jeju, Korea, Republic of. IEEE.

Pang, S., Kent, D., Cai, X., Al-Qassab, H., Morris, D., and Radha, H. (2018). 3d scan registration based localization for autonomous vehicles - a comparison of ndt and icp under realistic conditions. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5.

Pomerleau, F., Colas, F., Siegwart, R., et al. (2015). A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104.

Reijgwart, V., Cadena, C., Siegwart, R., and Ott, L. (2023). Efficient volumetric mapping of multi-scale environments using wavelet-based compression. In *Robotics: Science and Systems XIX*. Robotics: Science and Systems Foundation.

Ruan, J., Li, B., Wang, Y., and Sun, Y. (2023). SLAMesh: Real-time LiDAR Simultaneous Localization and Meshing. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3546–3552, London, United Kingdom. IEEE.

Saarinen, J., Andreasson, H., Stoyanov, T., Ala-Luhtala, J., and Lilienthal, A. J. (2013). Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping. In *2013 IEEE International Conference on Robotics and Automation*, pages 2233–2238, Karlsruhe, Germany. IEEE.

Schreier, M., Willert, V., and Adamy, J. (2015). Compact representation of dynamic driving environments for adas by parametric free space and dynamic object maps. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):367–384.

Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Robotics: Science and Systems V*. Robotics: Science and Systems Foundation.

Vespa, E., Funk, N., Kelly, P. H. J., and Leutenegger, S. (2019). Adaptive-resolution octree-based volumetric SLAM. In *International Conference on 3D Vision (3DV)*, pages 654–662.

Vespa, E., Nikolov, N., Grimm, M., Nardi, L., Kelly, P. H. J., and Leutenegger, S. (2018). Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151.

Vizzo, I., Chen, X., Chebrolu, N., Behley, J., and Stachniss, C. (2021). Poisson Surface Reconstruction for LiDAR Odometry and Mapping. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5624–5630, Xi'an, China. IEEE.

Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., and Stachniss, C. (2023). KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036.

Wang, K., Gao, F., and Shen, S. (2019). Real-time Scalable Dense Surfel Mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6919–6925, Montreal, QC, Canada. IEEE.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., and Burgard, W. (2010). OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems. In *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, volume 2, page 3.

Xu, M., Lin, S., Wang, J., and Chen, Z. (2023). A LiDAR SLAM System With Geometry Feature Group-Based Stable Feature Selection and Three-Stage Loop Closure Optimization. *IEEE Transactions on Instrumentation and Measurement*, 72:1–10.

Yguel, M., Aycard, O., and Laugier, C. (2006). Wavelet Occupancy Grids: A Method for Compact Map Building. In Corke, P. and Sukkariah, S., editors, *Field and Service Robotics*, volume 25, pages 219–230. Springer Berlin Heidelberg. Series Title: Springer Tracts in Advanced Robotics.

Young, M., Pretty, C., McCulloch, J., and Green, R. (2021). Sparse point cloud registration and aggregation with mesh-based generalized iterative closest point. *Journal of Field Robotics*, 38(8):1078–1091.

Zhang, J. and Singh, S. (2014). LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation.

Zhong, X., Pan, Y., Behley, J., and Stachniss, C. (2023). SHINE-Mapping: Large-Scale 3D Mapping Using Sparse Hierarchical Implicit Neural Representations. arXiv:2210.02299 [cs].

Zhu, Y., Zheng, X., and Zhu, J. (2024). Mesh-LOAM: Real-Time Mesh-Based LiDAR Odometry and Mapping. *IEEE Transactions on Intelligent Vehicles*, pages 1–12.

APPENDIX

Point Cloud Based Mapping

This section shows the trajectories of different voxel sizes by the subset of KiTTi sequneces.

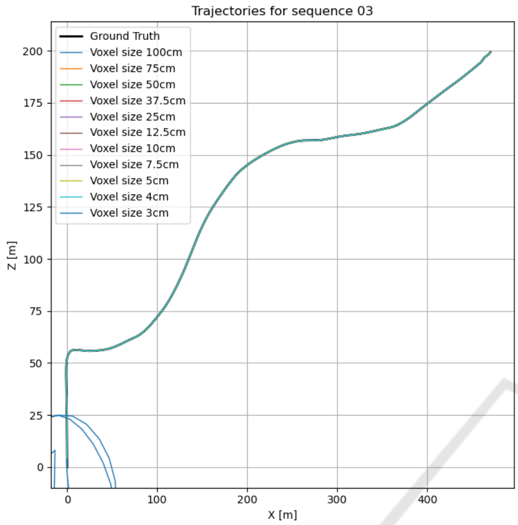

Figure 8: Trajectory for sequence 03 estimated using KISS-ICP, the vehicle moves from 0.0 to 480,200. The localization diverges for a voxel sizes of 3 cm.

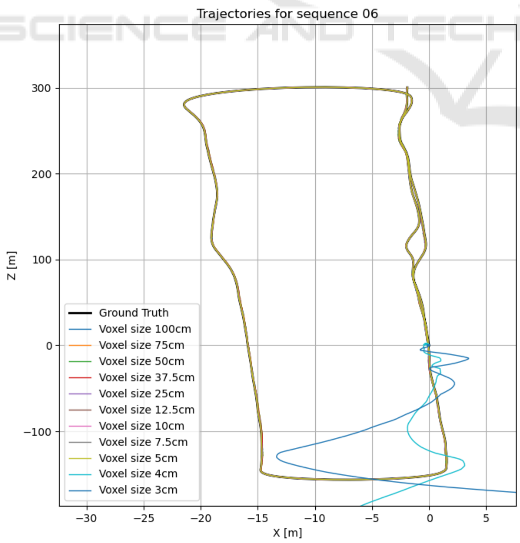

Figure 9: Trajectory for sequence 06 estimated using KISS-ICP, the vehicle moves in a loop from 0.0 to -3,300. The localization diverges below a voxel size 4cm.

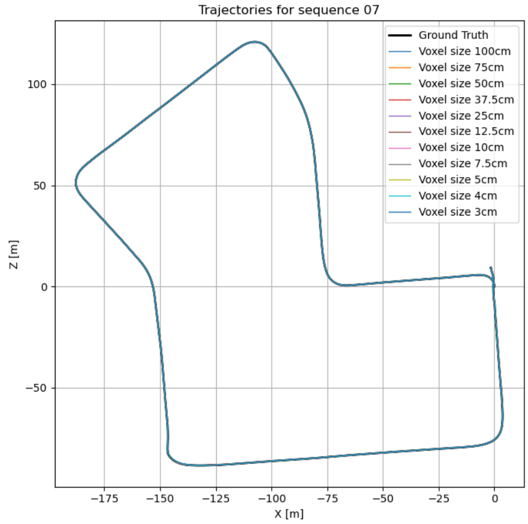

Figure 10: Trajectory for sequence 07 estimated using KISS-ICP, the vehicle moves in a loop from 0.0 to -5,10.

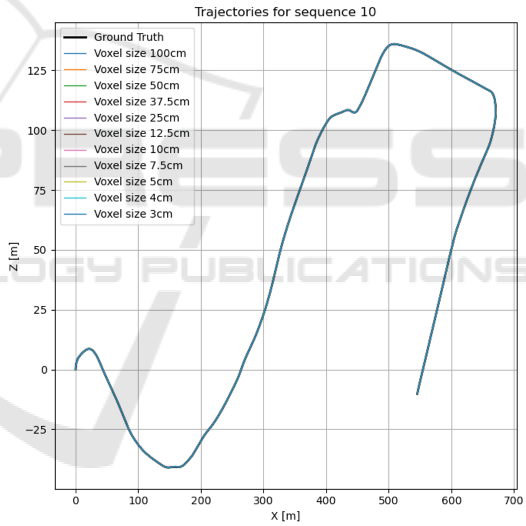

Figure 11: Trajectory for sequence 10 estimated using KISS-ICP, the vehicle moves in a loop from 0.0 to 550,-10.

Mesh Based Mapping

This section shows the trajectories of different depth values by the subset of KiTTi sequneces.

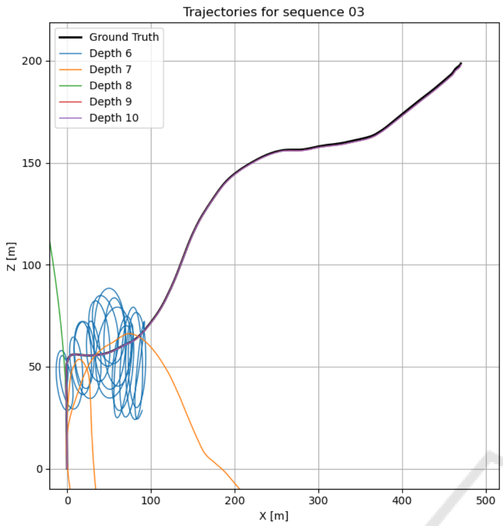

Figure 12: Trajectory for sequence 03 estimated using PUMA, the vehicle moves from 0.0 to 480,200. The localization diverges for a depth values of [6, 7, 8].

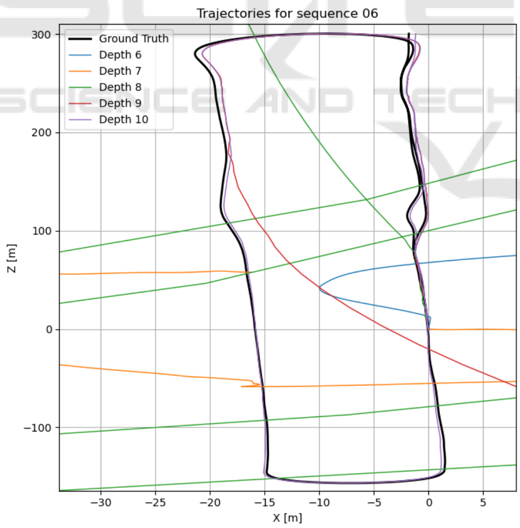

Figure 13: Trajectory for sequence 06 estimated using PUMA, the vehicle moves in a loop from 0.0 to -3,300. The localization diverges for a depth values of [6, 7, 8, 9].

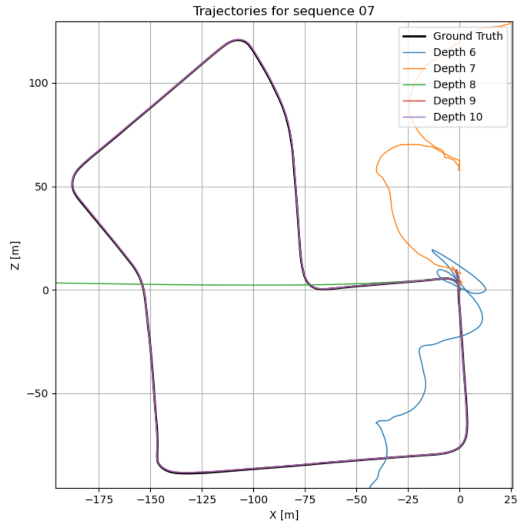

Figure 14: Trajectory for sequence 07 estimated using PUMA, the vehicle moves in a loop from 0.0 to -5,10. The localization diverges for a depth values of [6, 7, 8].

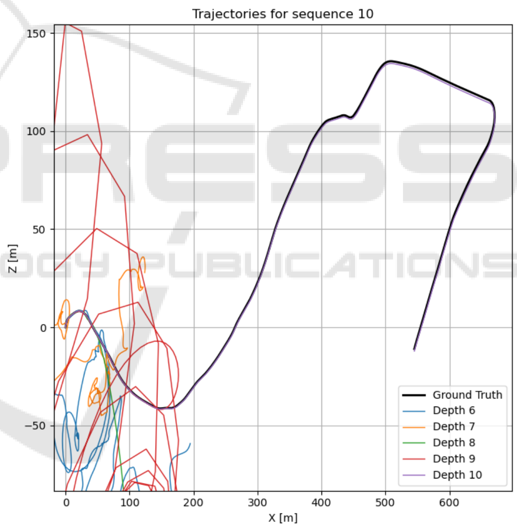

Figure 15: Trajectory for sequence 10 estimated using PUMA, the vehicle moves in a loop from 0.0 to 550,-10. The localization diverges for a depth values of [6, 7, 8, 9].