A Dynamic Object Removal and Reconstruction Algorithm for Point Clouds

Nagavarapu, Sarat Chandra ; Abraham, Anuj ; Selvaraj, Nithish Muthuchamy ; Dauwels, Justin

DOI
10.1109/SOLI60636.2023.10425733

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)

A Dynamic Object Removal and Reconstruction Algorithm for Point Clouds

Sarat Chandra Nagavarapu[1], Anuj Abraham[2], Nithish Muthuchamy Selvaraj[3] and Justin Dauwels[4]

[1]Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632
[2]Technology Innovation Institute, 9639 Masdar City, Abu Dhabi, United Arab Emirates
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore, 639798
[4]Department of Microelectronics, Faculty of EEMCS, TU Delft, Mekelweg 4, 2628 CD Delft, The Netherlands
Email: [1]Nagavarapu@i2r.a-star.edu.sg, [2]anuj.abraham@tii.ae, [3]ms.nithish@ntu.edu.sg, [4]J.H.G.Dauwels@tudelft.nl

2023 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI) | 979-8-3503-9600-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/SOLI60636.2023.10425733

***Abstract*—Autonomous vehicles (AV) are one of the greatest technological advancements of this decade and a giant leap in the transportation industry and mobile robotics. Autonomous vehicles face several major challenges in achieving higher levels of autonomy. One of these is to find a fast and reliable algorithm to process the sensor data so that the simultaneous localization and mapping (SLAM) algorithms run in real-time to achieve autonomous navigation. The major limitation of the SLAM algorithm, especially while building a map is to have static environmental features, i.e. without any dynamic or moving objects. To achieve this, our paper introduces a novel algorithm to remove dynamic objects from point cloud data. The algorithm focuses on identifying and removing dynamic objects from sensor data, thereby creating a static scene suitable for traditional SLAM algorithms. Simulations conducted on the benchmark dataset demonstrate the algorithm's efficacy in successfully eliminating dynamic objects and reconstructing a stable static scene.**

***Index Terms*—point clouds, autonomous vehicles, object removal, reconstruction, SLAM, LiDAR, mobile robots.**

I. Introduction

Autonomous Vehicle (AV) technology represents the promising future of the transportation sector. Leveraging cutting-edge technologies, AVs demonstrate the capacity to make superior decisions compared to their human counterparts. However, there have been instances where autonomous cars encountered failures, some of which resulted in tragic accidents. For instance, in a recent incident involving an Uber autonomous car, equipped with a safety driver, it failed to engage the brakes when a cyclist suddenly appeared in its path. This convergence of human and machine errors led to a loss of life.

The inherent challenge with this technology lies in the difficulty of pinpointing the precise causes of such failures in specific cases. Consequently, finding effective solutions to prevent such incidents in the future becomes a complex and intricate task. Identifying dynamic obstacles in the vicinity of an Autonomous Vehicle (AV) is of paramount importance to ensure safe operations and prevent accidents, thereby safeguarding human lives. The ability to detect and respond to dynamic elements such as pedestrians, cyclists, and other vehicles in real time is crucial for maintaining public trust in autonomous vehicle technology. Proactive obstacle identification not only enhances road safety but also contributes to the harmonious integration of autonomous vehicles into our daily lives, minimizing disruptions and potential hazards [1].

While the deployment of AVs at present would likely result in a reduction in road accidents, the overarching goal of this groundbreaking technology is to eventually eradicate road-related fatalities, ensuring an entirely safe environment for humans. The fundamental principle behind AV development aligns with the vision that machines should never be the cause of human harm, in line with Asimov's laws of robotics. Consequently, companies continue to refine and perfect AV technology, refraining from premature deployment to ensure it meets the highest safety standards and uphold this critical objective for the future [2].

The cornerstone of an Autonomous Vehicle's functionality lies in Simultaneous Localization and Mapping (SLAM). SLAM involves the vehicle's endeavor to map its surrounding environment and establish its precise location about various reference points, subsequently informing its decision-making process. Employing sensors, the vehicle scans and maps its surroundings, designating stationary elements as landmarks and determining its own position through distance calculations relative to these landmarks. Subsequently, a trajectory is formulated, where each point along the trajectory is defined in relation to these established landmarks.

Dynamic objects have long impeded real-time SLAM reliability. Multiple authors have proposed methods to segment moving objects while preserving the static scene, but these techniques invariably require a minimum of two separate images or point clouds. Jizhou Yan et al. [3] introduced sensor fusion, utilizing diverse car sensors. They detected moving objects using RGB images at varying time intervals and subsequently removed corresponding points in the sparse 3D Light Detection and Ranging (LiDAR) cloud based on the spatial data of the moving objects. A dynamic object aware LiDAR SLAM pipeline based on deep learned dynamic object filtering step was discussed [4]. This generic method applies to different dynamic objects, segmentation methods, and LiDAR SLAM algorithms.

Yuxiang Sun et al. [5] employs a distinct approach by transforming point clouds into RGB images and depth maps. They create a dynamic object mask using RGB and depth images from two different time frames and subsequently eliminate corresponding dynamic object points. For factory environment monitoring, [6] utilizes a voxel-based approach to LiDAR data. This algorithm employs a voxel grid data structure and marks voxels as dynamic through ray intersection tests, without necessitating ego-motion estimation, 3D object recognition, or tracking.

Jian Tang et al. [7] introduced the Likelihood Grid Voting (LGV) method for dynamic object removal, involving voting on grid occupancy. Grids with fewer votes are identified as dynamic objects and subsequently removed. They validated the algorithm in real-time on a UGV developed by them. An innovative approach using synthesized optical flow was introduced in [8]. They leverage Stereo Odometry to estimate camera motion and pixel disparity, enabling the calculation of synthesized optical flow over the same pixel space. By utilizing depth and image intensity, they eliminate regions of inconsistency, corresponding to dynamic objects. Krystof Litomisky et al. [9] use correspondences to compute the displacement matrix, with which the points of moving objects are identified. The points are then removed to create a static scene. Moreover, a robust method for removing dynamic objects was presented by using the occupancy octree map to create a clean point cloud, but not scalable for large-scale outdoor maps [10].

A straightforward dynamic object removal method, presented in [11], involves the use of two RGB-D images. This approach employs image differencing along with ego-motion compensation to detect changes. Subsequently, object tracking is performed using a particle filter, and vector quantization is applied for dynamic object removal. Tanwei Zhang et al. [12] introduce the Mean Axis Descriptor for moving object characterization. This descriptor, in combination with sensor data, facilitates the identification of dynamic objects. An additional benefit of this algorithm is its capability to rectify distortion caused by object motion. A method for efficiently eliminating moving objects from point clouds in autonomous driving scenarios was introduced [13]. The authors in [13] use the SemanticKITTI dataset and follow a similar idea to the Octomap 3D occupancy grid mapping approach to filter the moving objects.

Canbe Yin et al. [14] utilize two RGB-D images from depth cameras to perform image differencing, followed by a thresholding operation to detect dynamic objects. They remove dynamic objects by establishing a correspondence between candidate pixels and the point cloud cluster. In contrast, Yuxiang Sun et al. [15] presented a method that detects moving objects through dense pixel matching. Initially, they remove the plane, then, after identifying moving objects, create a mask to eliminate them. Importantly, this approach does not rely on prior information about the visual appearance of the objects.

Johannes Schauer et al. [16] introduce a method that involves traversing the point cloud into a voxel grid and explicitly determining the volumetric grid occupancy to detect dynamic objects. Their work also introduces the concept of point shadows. Krystof Litomisky et al. [9] also present a similar algorithm that shares the segmentation step before motion detection. Their approach employs correspondences to calculate a displacement matrix, allowing the identification and removal of points belonging to moving objects, ultimately creating a static scene.

Each of the outlined approaches is meticulously tailored to serve distinct application domains, spanning UAVs, factory monitoring systems, indoor mapping platforms, and robotic contexts. In a parallel vein, the algorithm delineated in this research paper has been intricately designed with an acute focus on its application within the sphere of AVs.

The primary contributions of this research paper encompass the following key aspects: 1) Implementation of a k-nearest neighbors method for point cloud outlier identification and removal. 2) Utilization of Random Sample Consensus (RANSAC) and Euclidean cluster extraction approaches for preprocessing the point cloud, for plane removal and object segmentation respectively. 3) Dynamic object detection in 3D point clouds using Octrees and the reconstruction of a static scene from the input point cloud data.

The subsequent sections of the paper are structured as follows: In Section II, we provide a brief overview of autonomous vehicle navigation concepts, challenges in autonomous vehicle (AV) operations, the role of LiDARs in AV navigation, and the concept of simultaneous localization and mapping (SLAM). Section III outlines the proposed algorithm for dynamic obstacle removal and static scene reconstruction from 3D point cloud data. Section IV offers detailed insights into simulation results and a performance comparison between the proposed algorithm and an existing technique. Finally, Section V presents conclusive remarks and outlines potential directions for future research.

II. Autonomous Navigation & Major Challenges

Autonomous vehicle has been a prominent buzzword for over a decade, poised to usher in a revolution not just in the automotive sector but also in the field of robotics. The AV market is estimated to be worth trillions of dollars, with the potential to disrupt both public transportation and personal mobility. This significant market potential has drawn the attention of major players in the automotive industry, who are engaged in fierce competition, channeling substantial funds into research and development to transform this technology into a tangible reality.

The concept of autonomous cars began to solidify and gain traction when Google entered the arena, creating significant ripples within the tech community. Google's reputation for investing in groundbreaking technologies lent added credibility to the autonomous vehicle field. Fig.1 depicts the AnnieWAY autonomous car used by KITTI [18]. Consequently, numerous major players in the automotive industry, along with startups, have committed substantial resources to the research and development of autonomous technology. Their collective goal is to attain full vehicle autonomy, often referred to as Level 5 automation in driving, with some companies like Tesla even pushing the envelope by introducing semi-autonomous features in their vehicles and deploying them in real-world road conditions.

Fig. 1. The AnnieWAY Autonomous System Platform used by KITTI [18].

Challenges in autonomous navigation encompass sensor reliability, real-time data processing, mapping and localization, dynamic object detection, path planning, cybersecurity, legal complexities, human-AV interaction, urban navigation, and public acceptance. These challenges involve ensuring sensor accuracy, managing data efficiently, mapping and localization, detecting dynamic objects, path planning, addressing cybersecurity risks, dealing with legal and regulatory issues, optimizing human-autonomous vehicle interaction, navigating complex urban environments, and gaining public trust.

A. Autonomous Vehicle: Sensors

To operate effectively, an AV must have a comprehensive view of its environment, capturing all relevant information from its surroundings. To achieve this, AVs employ a diverse array of sensors. Autonomous vehicles are equipped with a variety of sensors for environment perception. These include LiDAR for 3D mapping, radar for object detection, cameras for visual data, ultrasonic sensors for proximity detection, IMUs for motion tracking, GPS for accurate positioning, wheel encoders for speed estimation, thermal sensors for night vision, and microphones for auditory cues [20]. The combined data from these sensors is crucial for AVs to navigate safely and make real-time decisions on the road.

Among these sensors, LiDAR holds significant importance, as it provides a detailed 3D point cloud of the surrounding environment for autonomous navigation. LiDAR is the instrument of choice for capturing point clouds due to its exceptional accuracy, compact form factor, and suitability for robotics and autonomous vehicles, surpassing other options. LiDAR systems typically include a scanning laser that emits laser pulses at regular intervals, and a photo-detector records the time it takes for these pulses to bounce back after hitting objects. This data forms a point cloud, representing the environment with precise depth information. LiDAR technology is a fundamental component of autonomous vehicles, playing a pivotal role in their safe and accurate navigation. Fig.2 depicts a typical LiDAR sensor.

Fig. 2. A Typical LiDAR System - Velodyne Scanner [19].

B. Point Clouds

Point clouds are collections of points in a spatial coordinate system, with commonly used systems including spherical, cylindrical, and Cartesian coordinates. However, Cartesian coordinates, utilizing three axes (x, y, and z), are the prevalent choice for representing point clouds. These point clouds serve as a spatial representation, akin to images and graphs, and are typically generated using instruments like stereo cameras, LiDAR, and 3D scanners.

To enhance the information captured, additional channels may be incorporated, such as RGB color data in depth cameras like the Microsoft Kinect, resulting in a six-channel representation (XYZRGB) for each point in space. The selection of the number of channels and coordinate system relies on specific application requirements. Point cloud representations find diverse applications in fields like robotics, geography, construction monitoring, and Computer-Aided Design (CAD), among others. Fig.3 illustrates a 3D point cloud obtained by a Velodyne LiDAR mounted on the roof of an autonomous vehicle.

For decades, images served as the primary means to represent spatial data, but the challenge of obtaining depth information persisted. This need for combined position and depth details led to the development of point clouds, which offer a more accurate and noise-resistant representation of 3D information. Unlike images, point clouds are scattered in a 3D space, providing multiple viewpoints of a scene and enabling different observations to reconstruct the same object. Various processing techniques for point clouds are available through open-source libraries, making them an efficient solution for spatial data representation.

SLAM is a fundamental component in AVs and mobile robots, serving as a critical prerequisite for enabling autonomous behavior and intelligence for navigation. The following section provides a concise overview of SLAM concepts.

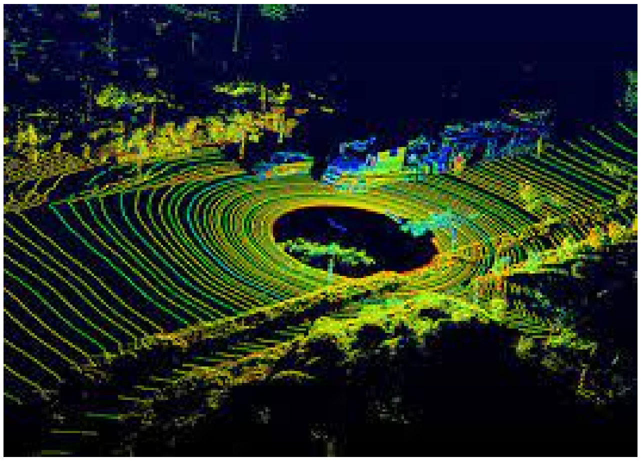


Fig. 3. An example of Point Cloud obtained from LiDAR [19].

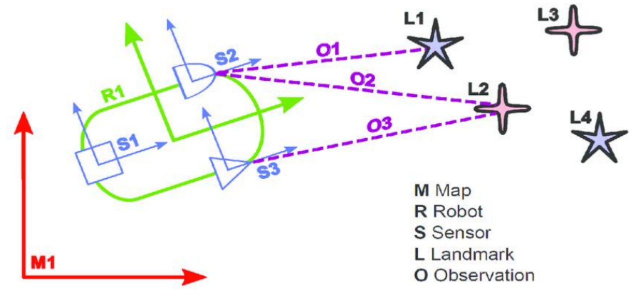


Fig. 4. An example of SLAM in three-wheeled mobile robot [20].

C. Simultaneous Localization And Mapping (SLAM)

Simultaneous Localization And Mapping (SLAM) is a process where a robot or autonomous vehicle creates a map of its environment while simultaneously determining its position on that map. It encompasses both mapping and localization operations carried out concurrently. A straightforward way to comprehend SLAM is by considering the questions a robot seeks to answer during its operation: 1) *what's around me?*, 2) *where am I?*, 3) *what should I do next?* These operations can be succinctly referred to as mapping, localization, and trajectory planning, respectively.

1) *Mapping*: Using LiDAR and cameras, the robot maps its environment by generating precise point clouds for depth information and capturing color and spatial data, either through sensor fusion or individual sensors.
2) *Localisation*: Within the map, the SLAM algorithm identifies stationary landmarks with low noise data and calculates the robot's position in relation to these landmarks. Examples of landmarks include lamp posts, medians, and traffic poles.
3) *Trajectory Planning*: The robot plans its trajectory, defining each point with respect to landmarks in terms of distance. It subsequently moves to these points at specific intervals, following the desired path when it accurately identifies and tracks the landmarks.

Problem in SLAM: Traditionally, SLAM algorithms presume stationary landmarks, but in real scenarios with dynamic objects, they can encounter challenges. The presence of moving entities, like people and pets, challenges the accuracy of generated maps. Accurate identification of stationary objects as landmarks is crucial for successful SLAM operation, as any misclassification of moving objects can lead to critical errors.

Distinguishing between moving and stationary objects is a complex task, often requiring higher-level intelligence. Even when identified, removing only moving objects while preserving stationary ones presents a challenge. To address this, a proposed algorithm aims to enable the use of conventional SLAM algorithms, reducing the need for human intervention when deploying mobile robots in real-time scenarios.

LiDAR Point Clouds for SLAM: SLAM can utilize point clouds for mapping, with LiDAR creating a point cloud representation of the environment. Objects within the point cloud serve as landmarks, enabling the robot to determine its position. In modern AV approaches, complete city maps, including landmarks, are pre-captured using 360-degree cameras and LiDAR for advanced navigation. In this approach, real-time point clouds from autonomous cars are compared with initial data for self-localization, demanding high-speed communication. However, on-board point cloud processing is preferred. To resolve the widespread challenge of distinguishing and eliminating moving objects from a point cloud while restoring a static scene, a novel algorithm is introduced.

III. Algorithm for Dynamic Object Removal

Dynamic objects can disrupt a robot's trajectory, rendering the resulting SLAM unreliable. In applications demanding precise trajectory tracing, the removal of dynamic objects is essential for enhancing SLAM reliability. The proposed algorithm effectively removes dynamic objects and is implemented through the Point Cloud Library (PCL). The proposed approach differs from existing methods by performing segmentation before dynamic object detection, resulting in faster algorithmic execution.

In the proposed method, two point clouds from different time frames are compared to identify moving objects and subsequently remove them. Unlike other techniques that rely on frame differencing, which only provides surface points of displaced objects, this method aims to capture all points within the object. This is achieved by segmenting all objects in both frames and comparing point clusters representing single objects. If the number of displaced points surpasses a threshold, the object is considered to have moved, and the entire cluster is discarded.

The algorithm is divided into two parts: the first part covers pre-processing and segmentation methods, while the second part details the techniques for moving object detection and the reconstruction of a stationary scene from the input point cloud. These operations utilize the Point Cloud Library and are implemented in C++. The process involves obtaining two point clouds, Cloud A and Cloud B, sampled from LiDAR data at different time instants (t_1 and t_2), which are then used to reconstruct the static scene.

Fig.5 illustrates the flow chart of the initial phase of the proposed algorithm, encompassing the removal of outliers and planes, followed by object segmentation.

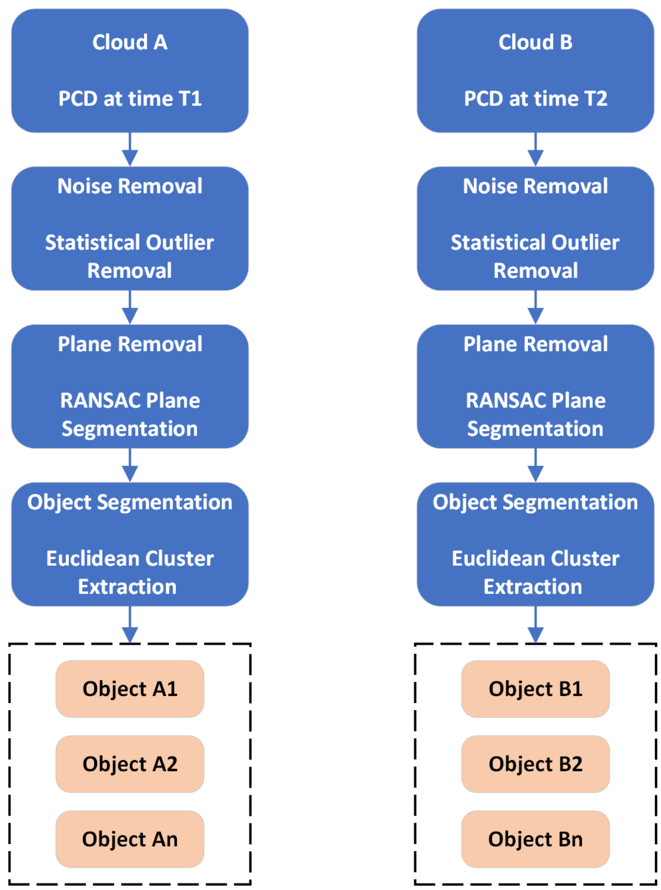


Fig. 5. Flow chart describing *noise removal* and *plane removal* operations.

Fig.6 illustrates the flow chart of the second phase of the proposed algorithm, explaining the methodology for detecting moving objects using Octrees and reconstructing a static scene.

The functional details of noise removal using the k-nearest neighbors approach, plane elimination using RANSAC, object segmentation employing Euclidean cluster extraction, and spatial change detection through Octrees are elaborated upon in the following sections.

A. Noise Removal

The point cloud obtained from the LiDAR contains too much noise which we refer to as outliers. Moreover, sparse points are also considered outliers. The sources of noise may be due to various reasons. It may arise from the photo-detector or the electronic circuitry in the processing system such as shot noise and thermal noise. The environment may also contain noise sources that may interfere with the wavelengths in which the laser beams operate. The presence of noise affects the

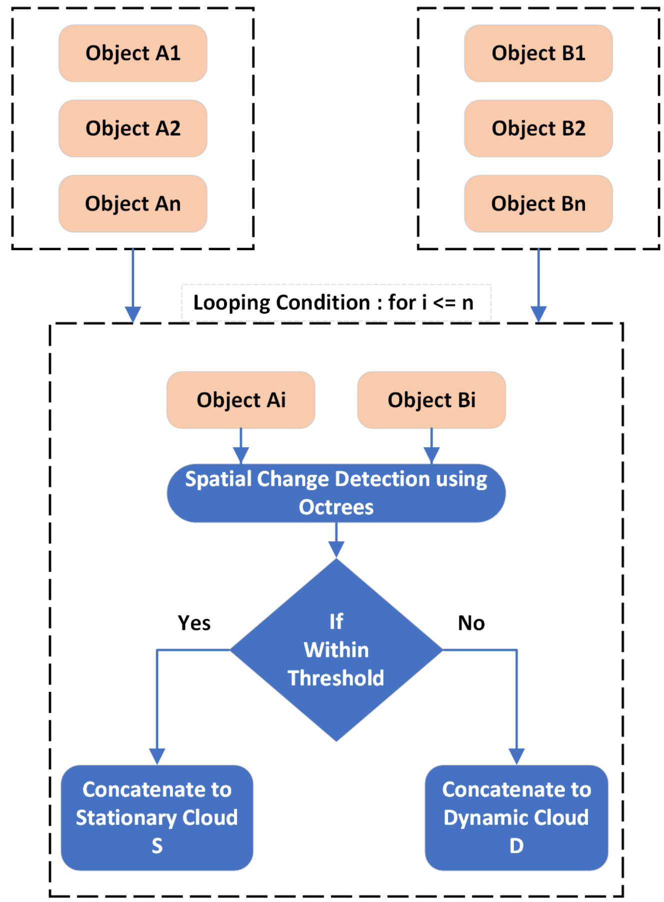


Fig. 6. Flow chart detailing *dynamic object detection* and *reconstruction* of a static scene.

operations done on the point clouds giving error-prone results and also the noise itself does not contain any information. Thus presence of noise is a waste of computational resources and removing it will yield efficient results.

The outlier removal in PCL is based on the Euclidean distance from the point to its neighbors. The K-nearest neighbor approach is used here. For every point in the input cloud, its distance to the nearest k neighbors is calculated. If the points are scattered and less dense the distance to the neighbours is high. Also, noise points are not in an orderly fashion with neighbors and hence their mean distances are also high. After performing the computation for all the points, a thresholding operation is performed in which the points whose mean distances are beyond a certain value are removed. As a result, the output cloud is noise-free and points are dense enough to process.

B. Plane Removal

In SLAM we aim to identify the static objects in the scene as a landmark and then try to plot a trajectory with the landmark as a reference and the landmark is not a plane. So removing all the points that represent a plane in the point cloud will significantly reduce the number of points and as a result, the processing requires less computation. The plane is identified and removed from the point clouds using the RANSAC segmentation process.

In PCL for the plane removal, the subset of points in the input cloud that closely resembles a plane are identified and then created as a separate cloud in the first iteration. Over the iterative process, all the points that are on a plane in the input cloud are added to the above cloud. As a result after the plane removal operation two clouds are created, one is the plane-removed input cloud and the other is the plane itself.

C. Object Segmentation

The object segmentation algorithm segments the input point cloud into individual objects which are nothing but individual smaller point clouds with high density. The algorithm starts with random points and starts to spread. Over time all the points are processed and the objects are segmented. The Euclidean Clustering works by first diving the unorganized point cloud into smaller parts or more specifically into octrees.

D. Spatial Change Detection

Spatial changes in the point cloud can be detected using octree representation. This is the most important operation in the algorithm and is the heart of the proposed algorithm in detecting dynamic objects. The octree-based spatial change detection is a more efficient way to detect the change compared to processing all of the points individually. This spatial change detection using octrees is done in a looping fashion in the proposed algorithm to detect the dynamic objects in LiDAR point clouds.

E. Motion Compensation

The aforementioned four essential operations collectively constitute the algorithm, leading to the removal of dynamic objects and the reconstruction of a static scene from an input point cloud. Nonetheless, a significant challenge remains unaddressed: orientation distortion in the point cloud resulting from the car's motion. When the car is stationary, it's relatively straightforward to identify and remove dynamic objects from the acquired point clouds. However, when the car is in motion, whether on a straight road or navigating a curve, all objects in the point clouds will exhibit apparent displacement between the two different time frames.

To counteract this displacement in point clouds caused by the car's motion, a motion compensation operation is executed. Utilizing data collected from the Inertial Measurement Unit (IMU) installed in the AV, the system calculates the displacement or rotation due to the vehicle's movement. These values are then used to create a homogeneous transformation matrix, which is employed to compensate for the car's motion within the point clouds.

While the proposed approach has several merits, it also comes with certain operational limitations. The following section outlines the strengths and operational constraints of this algorithm.

F. Merits & Limitations of the Proposed Algorithm

Merits: The advantages of the proposed algorithm are: 1) Faster Execution: It simplifies architecture by using a segmentation-first approach, eliminating complex mask generation. 2) Faster Processing: With reduced computational demands, it ensures faster processing, vital for resource-efficient systems like Autonomous Vehicles (AVs, where it serves as a pre-processing stage for SLAM.

Limitations: The algorithm excels in a pre-processing role but has limitations: 1) Closer Time Intervals: It falters with larger time gaps between samplings, potentially misidentifying objects due to significant displacements. 2) High Frequency of Operations: Optimal performance requires frequent data sampling, with cluttered, fast-paced environments benefiting more from complex methods operating at a lower rate.

The following section presents the simulation results obtained by applying the proposed algorithm to a benchmark dataset.

IV. Simulation Results

The proposed algorithm is implemented using the Point Cloud Library (PCL) in C++ on a computer running a Linux operating system. Given the relatively low computational requirements of this operation, it can be executed on a standard personal computer. The KITTI dataset [17] has established itself as the gold standard for autonomous vehicle (AV) research, serving as a benchmark for testing systems and assessing research progress within AV communities. To evaluate the performance of the proposed algorithm, it was tested using the KITTI benchmark dataset [17], which features real-time LiDAR data collected from autonomous cars navigating the streets of Karlsruhe, Germany. Furthermore, the algorithm's performance is benchmarked against a previously established dynamic object removal technique as outlined in [9] from the literature.

The accuracy calculation was specifically performed for objects with denser point distributions and excluded sparse points. This choice was made because in the LiDAR point cloud data, as the distance from the sensor increases, the points become increasingly scattered, and their resolution is insufficient for effective algorithm processing, ultimately leading to failure. Furthermore, in the context of SLAM, our primary focus is on nearby objects, which may serve as potential landmarks, and objects situated at a considerable distance are typically not of interest.

Tables I and II present the algorithm's performance with two distinct datasets, as illustrated in Fig.7, showcasing the point clouds of these datasets. The tables are divided into two sections. The *Near Distance* section pertains to objects with dense, closely located points near the autonomous car, featuring high point resolution and being more effective for processing. On the other hand, the *Far Distance* section pertains to objects represented by points located farther away from the car. *Ground Truth* represents the objects in the original point cloud, while *Processed Cloud* represents the objects after undergoing post-processing by the algorithm.

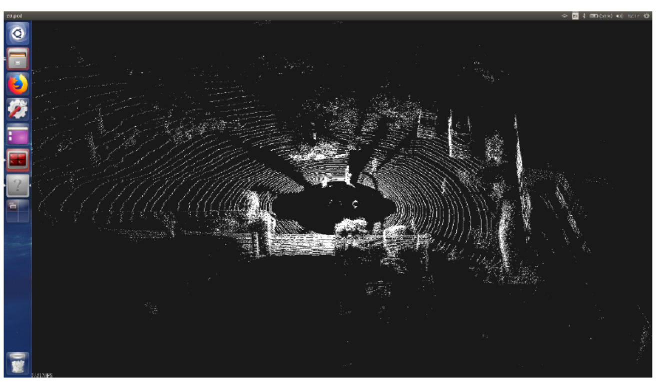
TABLE I
GROUND TRUTH AND PROCESSED CLOUD DATA FOR DATASET 1

Distance	Point Cloud Density	Object	Ground Truth			Processed Cloud Data		
			Stationary	Dynamic		Stationary	Dynamic	
				Slow Moving	Fast Moving		Slow Moving	Fast Moving
Near Distance	High (Dense)	Car	5	1	2	5	1	**0**
		Truck	1	–	–	1	–	–
		Lamp Post	10	–	–	**8**	–	–
Far Distance	Low (Sparse)	Buildings	4	–	–	**1**	–	–
		Trees/Bushes	2	–	–	**1**	–	–
		The rest of the points are too dispersed to be categorized as distinct objects.						

TABLE II
GROUND TRUTH AND PROCESSED CLOUD DATA FOR DATASET 2

Distance	Point Cloud Density	Object	Ground Truth			Processed Cloud Data		
			Stationary	Dynamic		Stationary	Dynamic	
				Slow Moving	Fast Moving		Slow Moving	Fast Moving
Near Distance	High (Dense)	Car	1	–	–	1	–	–
		Truck	1	–	–	1	–	–
		People	3	1	–	3	1	–
Far Distance	Low (Sparse)	Wall	4	–	–	**2**	–	–
		Trees/Bushes	8	–	–	**7**	–	–
		The rest of the points are too dispersed to be categorized as distinct objects.						

Dataset 1



Dataset 2

Fig. 7. Point clouds of two distinct KITTI datasets [17].

In Table I, the accuracy for dense points is 88.23%, whereas in Table II, it is 83.33%. Overall, the algorithm achieved an accuracy of 85.78%, while the authors of [9] attained an average accuracy of 85%. Although the difference in accuracy may not be notably high, the proposed approach attains quicker processing speeds in comparison to other methods. The decreased computational demands constitute a notable advantage in the context of SLAM computation.

V. CONCLUSIONS

Autonomous vehicular technology is considered to be one of the greatest technological advancements that humankind has achieved in the past decade. In autonomous navigation, LiDARs are commonly considered a crucial sensory component for a wide range of tasks. In this scenario, the proposed algorithm addresses dynamic object removal in 3D point clouds, offering particular applicability in the domains of robotics and smart urban mobility where LiDAR assumes a crucial role amid limited computational resources. The algorithm employs the k-nearest neighbors approach for noise removal in the point cloud data, plane elimination using RANSAC, object segmentation employing Euclidean cluster extraction, and spatial change detection through Octrees.

The algorithm attains an overall accuracy of 85.78%, signifying the successful retention of 85.78% of stationary points through a comparison of point clouds at two distinct time frames. While the algorithm showcases superior performance, its universal robustness is acknowledged to be context-dependent. Additionally, incorporating VoxelNet, a deep learning-based approach for point clouds, expands the research scope by enhancing 3D object recognition and presenting potential applications in dynamic object removal for future autonomous vehicles.

REFERENCES

[1] A. Abraham, S. C. Nagavarapu, S. Prasad, P. Vyas and L. K. Mathew, "Recent Trends in Autonomous Vehicle Validation Ensuring Road Safety with Emphasis on Learning Algorithms," 17th International Conference on Control, Automation, Robotics and Vi-

sion (ICARCV), Singapore, Singapore, 2022, pp. 397-404, doi: 10.1109/ICARCV57592.2022.10004304.

[2] Muhammad, U., Vyas, P., Abraham, A., Sundaram, A.R., Mehta, P.R., Dauwels, J., “An integrated simulator for testing and validation of autonomous vehicle applications with physics-based rendering sensors”, Proceeding in 26th ITS World Congress, Singapore, 21-25 October 2019, pp. 1-10. Available online at: https://hdl.handle.net/10356/152698

[3] J. Yan, D. Chen, H. Myeong, T. Shiratori and Y. Ma, ”Automatic Extraction of Moving Objects from Image and LIDAR Sequences,” 2nd International Conference on 3D Vision, Tokyo, Japan, 2014, pp. 673-680, doi: 10.1109/3DV.2014.94

[4] Patrick Pfreundschuh, Hubertus F.C. Hendrikx, Victor Reijgwart, Renaud Dubé, Roland Siegwart, and Andrei Cramariuc, ”Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data”, IEEE International Conference on Robotics and Automation (ICRA). IEEE Press, 2021, pp. 11641–11647.

[5] Sun, Y., Liu, M., Meng, M.Q., ”Invisibility: A moving-object removal approach for dynamic scene modelling using RGB-D camera”, IEEE International Conference on Robotics and Biomimetics (ROBIO), 2017, pp. 50-55.

[6] J. Schauer and A. Nüchter, ”Digitizing automotive production lines without interrupting assembly operations through an automatic voxel-based removal of moving objects,” 13th IEEE International Conference on Control & Automation (ICCA), Ohrid, Macedonia, 2017, pp. 701-706, doi: 10.1109/ICCA.2017.8003145.

[7] Yu, T., Liu, J., Chen, Y., & Tang, J., ”An Approach of Dynamic Object Removing for Indoor Mapping Based on UGV SLAM”, Sensors & Transducers, 190(7), 2015, pp. 40-46.

[8] O. K. Hamilton and T. P. Breckon, ”Generalized dynamic object removal for dense stereo vision based scene mapping using synthesized optical flow,” IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 3439-3443, doi: 10.1109/ICIP.2016.7532998.

[9] Litomisky Krystof, and Bir Bhanu. ”Removing moving objects from point cloud scenes.” In Advances in Depth Image Analysis and Applications: International Workshop, WDIA 2012, Tsukuba, Japan, November 11, 2012, Revised Selected and Invited Papers, pp. 50-58. Springer Berlin Heidelberg, 2013.

[10] Pagad Shishir, Agarwal Divya Narayanan Sathya, Rangan Kasturi, Kim Hyungjin, Yalla Ganesh., ”Robust Method for Removing Dynamic Objects from Point Clouds”, 2020, pp. 10765-10771. 10.1109/ICRA40945.2020.9197168.

[11] Yuxiang Sun, Ming Liu, Max QH Meng, ”Improving RGB-DSLAM in dynamic environments: A motion removal approach”, Journal on Robotics and Autonomous Systems, Elsevier, 2016.

[12] Tianwei Zhang and Yoshihiko Nakamura, ”Moving Humans Removal for Dynamic Environment Reconstruction from Slow-Scanning LIDAR Data”, 15th International Conference on Ubiquitous Robots (UR), 2018.

[13] Fu, H.; Xue, H.; Xie, G. MapCleaner: Efficiently Removing Moving Objects from Point Cloud Maps in Autonomous Driving Scenarios. Remote Sens. 2022, 14, 4496. https://doi.org/10.3390/rs14184496

[14] Canben Yin et.al, ”Removing Dynamic 3D Objects from Point Clouds of a Moving RGB-D Camera”, Proceeding of the 2015 IEEE International Conference on Information and Automation, 2015.

[15] Yuxiang Sun, Ming Liu, Max QH Meng, ”Motion removal for reliable RGBD SLAM in dynamic environments”, Journal on Robotics and Autonomous Systems, Elsevier, 2018.

[16] Johannes Schauer, Andreas Nüchter, ”Removing non-static objects from 3D laser scan data, ISPRS Journal of Photogrammetry and Remote Sensing”, Volume 143, September 2018, pp. 15-38.

[17] Available online at: http://www.cvlibs.net/datasets/kitti/

[18] Available online at: https://velodynelidar.com/

[19] Available online at: https://www.researchgate.net/publication/306097366_Development_of_a_Novel_Deliberate_Camera_Oscillation_System_to_Improve_Monocular_Visual_SLAM_Performance

[20] Y. Zhang, X. Shi, S. Zhang and A. Abraham, ”A XGBoost-Based Lane Change Prediction on Time Series Data Using Feature Engineering for Autopilot Vehicles,” in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, Oct. 2022, pp. 19187-19200, doi: 10.1109/TITS.2022.3170628.