

Cluster-based Dynamic Object Filtering via Egocentric Motion Detection for Building Static 3D Point Cloud Maps

Pengcheng Cao

Dept. of Mechanical Engineering
University of California San Diego
San Diego, USA
p5cao@ucsd.edu

Thomas Bewley

Dept. of Mechanical Engineering
University of California San Diego
San Diego, USA
tbewley@ucsd.edu

Falko Kuester

Dept. of Structural Engineering
University of California San Diego
San Diego, USA
fkuester@ucsd.edu

Abstract—In this work, we propose a lightweight dynamic object filtering algorithm for building LiDAR-based static point cloud maps in real-time. On one hand, we propose an egocentric motion detection method of using improved ICP to register 3D clusters and extract their poses and twists to identify dynamic objects. On the other hand, we connect the proposed dynamic object filter with LiDAR-based SLAM algorithms to build point cloud maps and validate the effectiveness of the proposed methodology on both our custom dataset and SemanticKITTI. We also compare the performance of the proposed method against state-of-the-art methods in terms of both filtering accuracy and processing time. As experimentally verified on SemanticKITTI, our method yields promising performance with relatively small time costs and therefore has great potential to be used as point cloud data source for a number of LiDAR-inertial-visual fusion mapping methods.

Keywords— Mapping, perception, LiDAR, SLAM.

I. INTRODUCTION

In recent years, the use of mobile robotic vehicles has witnessed significant growth in perception and mapping. From autonomous ground vehicles navigating complex urban [1]–[3] or unstructured environments [4]–[6] to drones charting unexplored terrains [7] [8], the fusion of perception and mapping capabilities has unleashed a new era of possibilities.

On the sensor side, 3D LiDAR technology plays an essential role in the field of mapping and perception in autonomous systems. Its ability to reconstruct dense 3D point clouds of the environment has paved the way for various applications. The popularity of LiDAR has also fostered the research in mapping algorithms, including the Iterative Closest Point (ICP) [9] [10], Simultaneous Localization and Mapping (SLAM) with LiDAR alone [11] [12] and fusing LiDAR with other sensors [13]–[15]. Among these, LiDAR-inertial odometry-based SLAM has the advantages of being able to mitigate accumulated drifts [1], relatively low computational cost [12], [14], [15], and maintain accuracy over long-distance travel [16] [17]. Therefore, LiDAR-inertial odometry is often used in the scenarios with impaired visual features or where long-term real-time stable motion estimation is required such as autonomous driving.

However, one of the significant challenges in utilizing LiDAR data for mapping and navigation is the presence of dynamic objects within the environment [18]. On one hand, during the point cloud registration, dynamic objects may introduce inconsistencies in static maps by occluding the static part of environment [19] and drifting away the odometry to cause misalignment when registering multiple scans into a single map [20]. On the other hand, regarding map

construction, dynamic objects can cause ghost trail effect and therefore leave residual or noise points to become part of the final generated static map [21]. To overcome this challenge, researchers have proposed a number of state-of-the-art algorithms, including ERASOR [21], Remover [22], and Peopleremover [23]. All of these methods, however, require a prior map to be created before filtering out dynamic objects points in the post-processing. There has been no outstanding findings or methods in removing dynamic objects on the fly when a prior map is generated known to the authors.

To fill this research gap, we develop a fast real-time dynamic object detection and filtering algorithm. The contributions of this work are three-fold. First, we give the problem definition of cleaning dynamic objects points while building the “prior” map and consider it as a component of LiDAR-Inertial-Visual fused mapping algorithm. Second, we propose an egocentric motion detection method of for point cloud clusters and estimate their poses and twists to identify dynamic objects. Last but not least, we connect the proposed filter with LiDAR-based SLAM algorithms to build point cloud maps and validate the effectiveness of the proposed methodology on both our custom dataset and SemanticKITTI [24].

II. METHODOLOGY

In this section, we are going to introduce the problem definition and workflow of the proposed cluster-based dynamic object filtering method. Note that the methods proposed in this work most likely apply to LiDAR sensors mounted on the ground vehicles or near-ground flying vehicles perceiving indoor or outdoor environments. Thus noise filtering for aerial-to-ground imaging or air-to-air perception is beyond the scope of this work.

A. Problem Definition

First we start with the definition of a generic point cloud mapping algorithm. Let $\mathcal{S}_{scan}^t = \{\mathbf{p}_1^t, \mathbf{p}_2^t, \mathbf{p}_3^t, \dots, \mathbf{p}_n^t\}$ be the set of all points in LiDAR’s current frame of scan, where each point with timestamp t is a vector of $\mathbf{p}_k^t = [x_k, y_k, z_k]^T$ if we assume other information like intensity and RGB values is not available. And let $\mathcal{S}_Q^t$ be the set of points in the keyframe or being queried at timestamp t . If we assume the beginning pose of the LiDAR corresponds to the origin and axes definition of the inertial frame, we can compute the SE(3) world-to-local transformation for this query ${}^W\mathbf{T}_Q^t$ at timestamp t . Therefore, for a mapping algorithm with raw point cloud streaming, the map $\mathcal{M}_{raw}$ can be obtained by:

$$\mathcal{M}_{raw} = \bigcup_{t \in \tau_Q} \{{}^W\mathbf{T}_Q^t \cdot \mathbf{p}_k^t \mid \mathbf{p}_k^t \in \mathcal{S}_Q^t, k = 1, 2, \dots, n\} \quad (1)$$

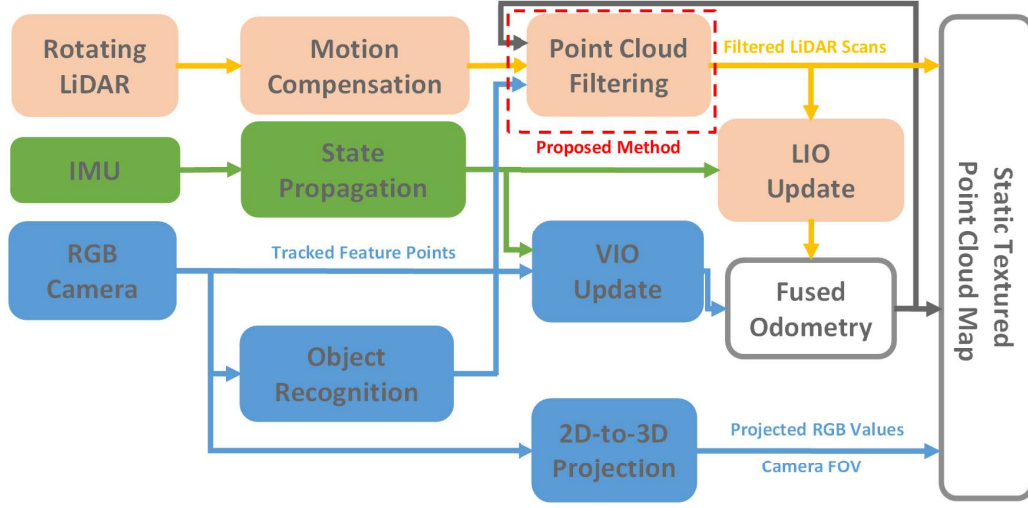


Fig. 1: LiDAR-inertial-visual fused SLAM system overview.

where τ_Q represents the set of timestamps when a query scan is obtained. Note that $\mathcal{M}_{raw}$ may contain dynamic objects and is mounted on the cartesian coordinate systems originated at the LiDAR's initial pose.

In post-processing algorithms like ERASOR [21], the problem formulation may be considered as removing the estimated dynamic object points after the $\mathcal{M}_{raw}$ is built, such as:

$$\hat{\mathcal{M}}_{static} = \mathcal{M}_{raw} - \bigcup_{t \in \tau_Q} {}^W \mathbf{T}_Q^t \odot \hat{\mathcal{S}}_{dyn}^t, \quad (2)$$

where $\hat{\mathcal{S}}_{dyn}^t$ represents the set of estimated dynamic object points at timestamp t , and $\odot$ here indicates point-wise multiplications similar to the operation in Eq. (1).

However, since we are attempting to filter out the dynamic points at the very time when the query scan is received, our problem is defined in a different way. We attempt to subtract the points directly from each query scan before unioning the query scans instead of subtracting the estimated dynamic objects after the raw map is built, such as:

$$\hat{\mathcal{M}}_{static} = \bigcup_{t \in \tau_Q} {}^W \mathbf{T}_Q^t \odot (\hat{\mathcal{S}}_Q^t - \hat{\mathcal{S}}_{dyn}^t). \quad (3)$$

Therefore, as the query or keyframe point clouds are filtered at the time they are achieved, it is made possible that we can build an estimated static point cloud map in real-time if the filters can have relative low computational time for each keyframe.

B. Potential Dynamic Objects

As we are proposing a dynamic object filtering algorithm mainly focus on near-ground objects, we observed a number of dynamic object clusters in various datasets and need to make reasonable assumptions before proposing the cluster-based filtering method. We discuss the outcome of observations and assumptions as follows.

Observation 1: The dynamic objects in both indoor and outdoor environments are most likely in contact with the ground plane. The categories of these objects include but are not limited to pedestrians, moving furniture, carts, bike and scooter riders, non-airborne animals, and vehicles etc.

Observation 2: The dynamic objects are most likely rigid bodies whose cross-section on XY-plane at each Z-level can be contained in convex hulls with the centroid of cross-section of this object inside its convex hull.

With the above mentioned statements, we can construct six possible cases for each of the detected object cluster:

- *Case 1:* A moving object cluster is in contact with the ground plane and is observed in both $\hat{\mathcal{S}}_{dyn}^t$ and $\hat{\mathcal{S}}_Q^t$, and therefore identified as a moving object. This is considered true positive (TP).
- *Case 2:* A static object cluster is in contact with the ground plane and is observed in $\hat{\mathcal{S}}_Q^t$ but not in $\hat{\mathcal{S}}_{dyn}^t$, and therefore identified as a static object. This is considered true negative (TN).
- *Case 3:* A static object cluster is in contact with the ground plane and is observed in $\hat{\mathcal{S}}_Q^t$ and $\hat{\mathcal{S}}_{dyn}^t$, and therefore identified as a moving object. This is considered false positive (FP).
- *Case 4:* A moving object cluster is in contact with the ground plane and is observed in $\hat{\mathcal{S}}_Q^t$ but not in $\hat{\mathcal{S}}_{dyn}^t$, and therefore identified as a static object. This is considered false negative (FN).
- *Case 5:* A static or moving object cluster is not in contact with the ground plane. This cluster is not considered in the scope of this work and therefore not admissible to either $\hat{\mathcal{S}}_Q^t$ or $\hat{\mathcal{S}}_{dyn}^t$.
- *Case 6:* A static or moving object cluster is in contact with the ground plane but not observed in either $\hat{\mathcal{S}}_Q^t$ or $\hat{\mathcal{S}}_{dyn}^t$. This may be caused by the flaw of clustering algorithm being used.

According to the derivation of error metrics in [21], it is possible that the number of static points falsely predicted as moving objects (FP) in a map can be much larger than correctly predicted moving objects (TP). Therefore, we adopt two novel quantitative metrics named as Rejection Rate (RR) and Preservation Rate (PR) as follows:

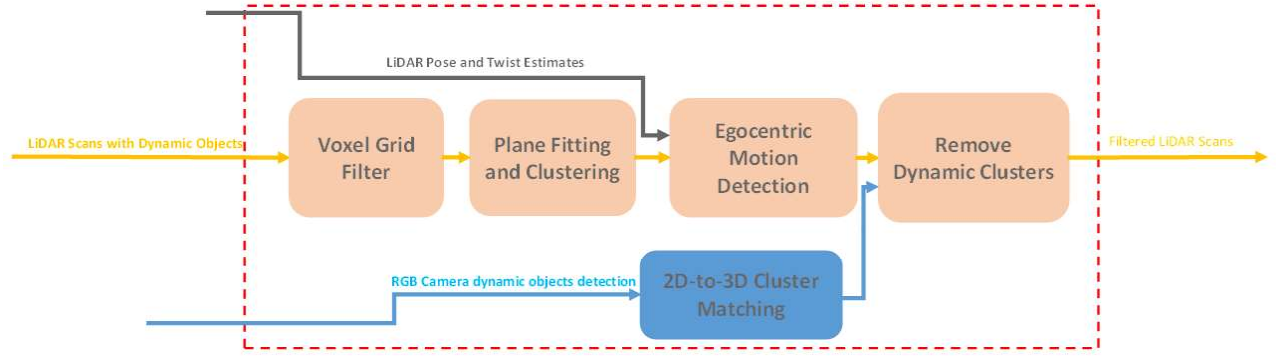


Fig. 2: Dynamic object filter design.

$$\begin{aligned}
 PR &= \frac{\# \text{ of preserved static points}}{\text{ground truth of static points}} \\
 &= \frac{TN + \text{ground plane}}{TN + FN + \text{ground plane}}, \\
 RR &= \frac{\# \text{ of removed dynamic points}}{\text{ground truth of dynamic points}} \\
 &= \frac{TP}{TP + FP}.
 \end{aligned} \tag{4}$$

C. Cluster-based Object Detection

Since we only consider objects connected to the ground plane, it is suitable to build a clustering pipeline for the environment once we can easily fit and extract the points on or close to the ground plane. Mathematically, a distinct cluster can be defined as a set of points C_i^t , and we also define another cluster C_j^t if all points $\mathbf{p}_j^t \in C_j^t$:

$$\min \|\mathbf{p}_j^t - \mathbf{p}_i^t\|_2 \geq d_{ij}^{th} \text{ for all } \mathbf{p}_i^t \in C_i^t, \tag{5}$$

where d_{ij}^{th} is the distance threshold between points in clusters C_i^t and C_j^t . In this work, we use a kd-tree representation-based nearest neighbors queries to partition keyframe LiDAR scans into clusters. The details of the Euclidean distance based clustering method can be found in [25].

D. Egocentric Motion Detection for Clusters

In LiDAR based SLAM algorithm, Since we have the estimated localization and self-motion streamed from odometry, we can compute the estimation of LiDAR keyframe pose ${}^W\hat{\mathbf{T}}_Q^t$ and corresponding twist $\hat{\zeta}_W^t$ at timestamp t . With the time sequence of ${}^W\hat{\mathbf{T}}_Q^t$ and $\hat{\zeta}_W^t$ characterizing LiDAR's ego-motion, one can estimate the motion of identified clusters in the environment as well.

In this work, we propose an adapted version of ICP-based algorithm for object motion detection as in Algorithm 1 depending on the time step length between two keyframes. By taking advantage of multiple processor cores onboard, it's possible to achieve near-real-time performance, especially with well-optimized implementations and capable hardware. However, for more complex environments or resource-constrained devices, real-time processing might be more challenging to achieve.

After obtaining the motion estimations of clusters at each time step, we can identify dynamic objects by inspecting the list of object to check their inconsistency with the inversed self-motion of the LiDAR query scans ${}^W\mathbf{T}_Q^{t-1}$. It is quite obvious that the static objects ought to have trivial twist estimation in the world frame, while dynamic object cluster can have varying twists estimation

Algorithm 1 Egocentric Motion Detection of Environmental Object Clusters in One Time Step

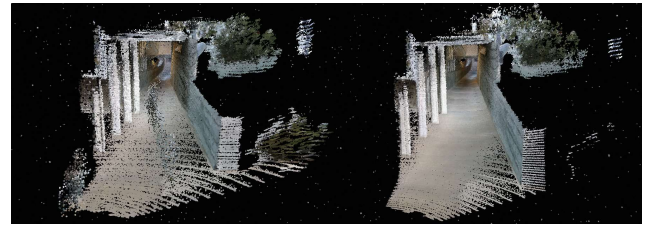
Require: Original query data S_Q^t at time $t - h$ and t

Require: LiDAR or fused odometry ${}^W\hat{\mathbf{T}}_Q^t$ and $\hat{\zeta}_W^t$ at time $t - h$ and t

Require: List of object clusters $\{C_i^t \mid i = 1, \dots, m\}$ at time $t - h$ and t , where m is the number of clusters

- 1: Init empty list of object poses $\{{}^W\hat{\mathbf{T}}_i^t \mid i = 1, \dots, m\}$
- 2: Init empty list of object twists $\{\hat{\zeta}_{i,W}^t \mid i = 1, \dots, m\}$
- 3: **for** each object cluster C_i^t **do**
- 4: Initialize transformation ${}^W\hat{\mathbf{T}}_{i,\text{init}}^t$ based on ${}^W\hat{\mathbf{T}}_{i,\text{refined}}^{t-h}$
- 5: Apply ICP to align points in C_i^t using ${}^W\hat{\mathbf{T}}_{i,\text{init}}^t$
- 6: Set ${}^W\hat{\mathbf{T}}_{i,\text{refined}}^t$ as the ICP-refined transformation
- 7: Replace ${}^W\hat{\mathbf{T}}_{i,\text{init}}^t$ with ${}^W\hat{\mathbf{T}}_{i,\text{refined}}^t$ in $\{{}^W\hat{\mathbf{T}}_i^t \mid i = 1, \dots, m\}$
- 8: Compute using poses of $t - h$ and t and Add the twist $\hat{\zeta}_{i,W}^t = (\mathbf{v}_i^t, \mathbf{w}_i^t)$ to $\{\hat{\zeta}_{i,W}^t \mid i = 1, \dots, m\}$
- 9: **end for**
- 10: **return** $\{{}^W\hat{\mathbf{T}}_i^t \mid i = 1, \dots, m\}, \{\hat{\zeta}_{i,W}^t \mid i = 1, \dots, m\}$

and changing poses the world frame. We use this principle to filter out the dynamic object clusters by setting the threshold for twist variations.



(a) Before filtering.

(b) After filtering.

Fig. 3: Ghost trail removal on reconstructed UCSD Calit2 tunnel.

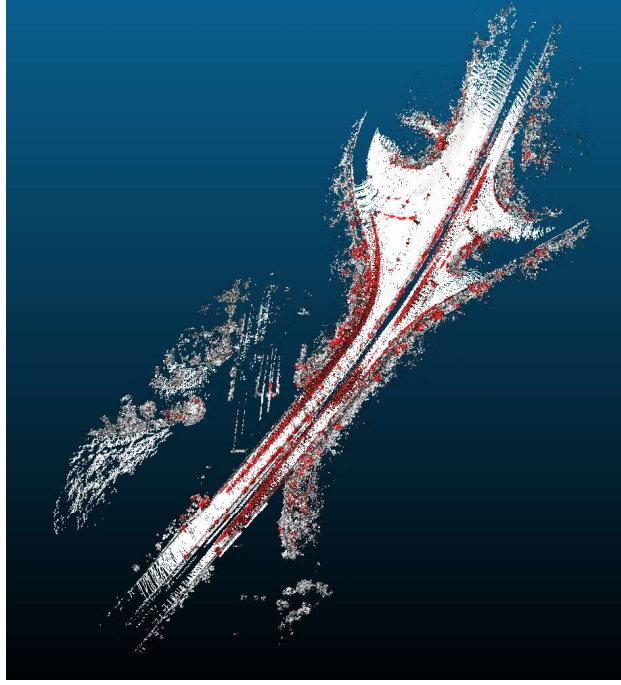


Fig. 4: Dynamic objects removal (red) on SemanticKITTI Sequence 01.

III. RESULTS AND DISCUSSION

In this section, we present the results of evaluating our dynamic object filtering algorithm on both our custom rosbag SLAM dataset and the publicly available SemanticKITTI sequences [24]. We assess the effectiveness of the algorithm in identifying and filtering out dynamic objects from LiDAR scans.

A. Custom dataset

We conducted preliminary case study tests using our custom research robotic platform DamBot-Mini. The dataset is simulating a tunnel environment with a mix of dynamic and static objects. Therefore, we apply our dynamic object filtering algorithm to perform LiDAR-inertial-visual mapping from the LiDAR scans to remove the "ghost trails" left by pedestrians from the static environment. In Fig.3, we can tell the ghost trails have been removed and the precision of the map is improved from a qualitative perspective.

B. SemanticKITTI

We extend our evaluation to the widely used SemanticKITTI dataset using a segment of Sequence 01 (frame no. 150 to 250). We convert the The algorithm's performance was assessed qualitatively by visually comparing the detected dynamic object regions with the point cloud sequences. Fig. 4 illustrate the effectiveness of our dynamic object filter applied to the sequence of selected frames of Sequence 01. We also apply the change detection method using CloudCompare's M3C2 algorithm [26] and mark the removed points in red as shown in Fig. 4. We also perform comparisons of different dynamic objects removing methods on the same sequence using OctoMap [27], Peopleremover [23], Removert [22], and ERASOR [21] as shown in Table I. Our proposed methodology has the shortest runtime compared to the other SOTA dynamic points cleaning methods.

TABLE I: Quantitative Comparisons on SemanticKITTI Sequence 01 (102 frames, bag time 10.4s).

Method	RR [%]	PR [%]	Runtime [s]
OctoMap[4]	99.863	20.777	120.254
People-remover[5]	93.116	36.349	112.412
Removert[6]	57.077	95.815	95.131
ERASOR[1]	95.383	91.487	17.866
Proposed	94.231	93.427	10.706

IV. CONCLUSION

In conclusion, this paper attempts to address a critical challenge in point cloud mapping, namely removing dynamic objects points from query LiDAR scans. We take a step forward by proposing a novel and lightweight dynamic object filtering algorithm, designed to facilitate the real-time generation of LiDAR-based static point cloud maps.

The method outlined in this paper holds great potential as a reliable point cloud data source for diverse LiDAR-inertial-visual fusion mapping approaches. Future work may involve studying the pros and cons of deep learning based, transformer based, and distanced cluster based methods for identifying dynamic objects.

ACKNOWLEDGMENTS

This publication is based on work supported by the US Army Corps of Engineers under research Cooperative Agreement W912HZ-17-2-0024 and under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and by the LLNL-LDRD Program under Project No. 20-SI-005. We thank all collaborators at the Qualcomm Institute and the Contextual Robotics Institute UC San Diego, as well as all other contributors to ideas, suggestions and comments. Opinions, findings, and conclusions from this study are those of the authors and do not necessarily reflect the opinions of the research sponsors.

REFERENCES

- [1] X. Xia, Z. Meng, X. Han, H. Li, T. Tsukiji, R. Xu, Z. Zheng, and J. Ma, "An automated driving systems data acquisition and analytics platform," *Transportation research part C: emerging technologies*, vol. 151, p. 104120, 2023.
- [2] J. Kocić, N. Jovičić, and V. Drndarević, "Sensors and sensor fusion in autonomous vehicles," in *2018 26th Telecommunications Forum (TELFOR)*. IEEE, 2018, pp. 420–425.
- [3] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep learning sensor fusion for autonomous vehicle perception and localization: A review," *Sensors*, vol. 20, no. 15, p. 4220, 2020.
- [4] J. Alberts, D. Edwards, T. Soule, M. Anderson, and M. O'Rourke, "Autonomous navigation of an unmanned ground vehicle in unstructured forest terrain," in *2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems (LAB-RS)*. IEEE, 2008, pp. 103–108.
- [5] P. Cao, J. Strawson, X. Zhu, E. Zhou, C. Lazar, D. Meyer, Z. Zheng, T. Bewley, and F. Kuester, "Beaglerover: An open-source 3d-printable robotic platform for engineering education and research," in *AIAA SCITECH 2022 Forum*, 2022, p. 1914.
- [6] S. Shimoda, Y. Kuroda, and K. Iagnemma, "Potential field navigation of high speed unmanned ground vehicles on uneven terrain," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2828–2833.
- [7] M. Bryson and S. Sukkarieh, "Co-operative localisation and mapping for multiple uavs in unknown environments," in *2007 IEEE aerospace conference*. IEEE, 2007, pp. 1–12.

- [8] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.
- [9] M. L. Tazir, T. Gokhool, P. Checchin, L. Malaterre, and L. Trassoudaine, "Cicp: Cluster iterative closest point for sparse–dense point cloud registration," *Robotics and Autonomous Systems*, vol. 108, pp. 66–86, 2018.
- [10] Y.-J. Lee and J.-B. Song, "Three-dimensional iterative closest point-based outdoor slam using terrain classification," *Intelligent Service Robotics*, vol. 4, pp. 147–158, 2011.
- [11] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.
- [12] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct lidar odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022.
- [13] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.
- [14] W. Xu and F. Zhang, "Fast-lío: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.
- [15] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lío2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.
- [16] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 5692–5698.
- [17] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-lidar-inertial odometry and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5167–5174, 2021.
- [18] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 4372–4378.
- [19] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1557–1563.
- [20] S. Yang, X. Zhu, X. Nian, L. Feng, X. Qu, and T. Ma, "A robust pose graph approach for city scale lidar mapping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1175–1182.
- [21] H. Lim, S. Hwang, and H. Myung, "Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021.
- [22] G. Kim and A. Kim, "Remove, then revert: Static point cloud map construction using multiresolution range images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10758–10765.
- [23] J. Schauer and A. Nüchter, "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid," *IEEE robotics and automation letters*, vol. 3, no. 3, pp. 1679–1686, 2018.
- [24] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.
- [25] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, pp. 345–348, 2010.
- [26] D. Lague, N. Brodu, and J. Leroux, "Accurate 3d comparison of complex topography with terrestrial laser scanner: Application to the rangitikei canyon (nz)," *ISPRS journal of photogrammetry and remote sensing*, vol. 82, pp. 10–26, 2013.
- [27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.