Enhancing LiDAR-Based Pole Localization for Autonomous Systems through Iterative Closest Point Integration

Tre' R. Jeter
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, Florida 32611, USA
t.jeter@ufl.edu

ABSTRACT

Improving localization algorithms using LiDAR data is paramount for advancing the accuracy and reliability of environmental mapping and navigation systems. LiDAR sensors provide high-resolution 3D point cloud data, offering detailed insights into the surroundings. In this study, the SemanticKITTI dataset served as the testing ground for pole localization, a crucial aspect of urban mapping and infrastructure analysis. Leveraging a LiDAR-based pole detection algorithm, the study aimed to precisely estimate the positions of vertical structures. Subsequently, the Iterative Closest Point (ICP) algorithm was applied to refine the alignment of localized maps with the global map, enhancing the overall accuracy of the localization results. Evaluation metrics, including Precision, Recall, and F1-score, were employed to systematically assess the performance improvements achieved through the integration of ICP. This research contributes to the ongoing efforts to enhance the robustness and applicability of LiDAR-based localization methods in diverse real-world scenarios.

KEYWORDS

LiDAR, Pole Localization, Iterative Closest Point, Autonomous Systems

1 INTRODUCTION

In the realm of autonomous systems, a transformative era is underway, with self-driving cars, autonomous ground robots, and drones seamlessly integrating into various facets of our lives, from revolutionizing transportation to undertaking complex tasks such as terrain exploration and contamination handling. The success of these intricate applications hinges upon the ability of autonomous entities to precisely ascertain their position in relation to the environment they navigate. Achieving centimeter-level precision in this estimation is paramount for ensuring the safe and effective operation of these cutting-edge technologies.

Recent revelations, however, have brought to light a new dimension of challenges. Studies like [2] have shown that LiDAR-based perception—the cornerstone of accurate spatial awareness in autonomous systems—is susceptible to manipulation through electromagnetic interference (EMI). As these technologies become more integral to our daily lives, understanding and mitigating vulnerabilities like those exposed by EMI injections are critical.

This project, in response to these emerging challenges, sets out to comprehensively characterize and evaluate the performance of a state-of-the-art path following technique prior to facing attacks. Specifically, this report delves into the implementation of the **O**nline **R**ange **I**mage-based **P**ole **E**xtractor (ORIPE) algorithm, a pole localization method, coupled with Iterative Closest Point (ICP) matching of the SemanticKITTI dataset [1]. The focus is on assessing the baseline performance of pole localization and then improving this performance by further aligning the local maps and global map through ICP matching. By undertaking this endeavor, the project aims to provide insights on increasing the robustness of LiDAR-based path following algorithms, contributing to the ongoing discourse on securing the autonomy of these systems in real-world scenarios.

Contributions. This project contributes the following:

- Pole localization and ICP Integration: This project applies the ORIPE pole localization technique to LiDAR data, particularly in the context of the SemanticKITTI dataset. Leveraging this foundation, the Iterative Closest Point (ICP) algorithm is integrated to enhance the alignment of local maps with the global map.
- Comprehensive Metric Assessment: The study incorporates a multifaceted evaluation approach, measuring the performance of the localization algorithms through metrics such as Precision, Recall, and F1-score. This comprehensive assessment provides a nuanced understanding of the algorithms' efficacy.

CIS6930: Cyber-physical System Security 1(1), 1–7

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- Real-world Applicability: By addressing challenges and improvements in LiDAR-based localization, this project contributes valuable insights into the robustness of autonomous systems. The findings have practical implications for enhancing the security and performance of these systems in real-world scenarios.

Organization. The rest of the report is organized as follows: Section 2 covers the background and related work to LiDAR-based research, pole localization, and integration of the Iterative Closest Point (ICP) algorithm. Section 3 describes the methodology used to carry out the proposed experiments. Section 4 illustrates the experimental results. Section 5 is an in-depth analysis of the experimental results. Section 6 is an open discussion. Section 7 concludes the report.

2 BACKGROUND AND RELATED WORK

2.1 LiDAR-based Perception for Autonomous Systems

LiDAR technology plays a pivotal role in the precise spatial perception of autonomous systems, relying on laser beams to create detailed 3D maps of the environment. The demand for centimeter-level accuracy in localization is critical for the safe and efficient operation of autonomous systems. Rieken et al. [14] introduced the *Stadtpilot* frameworks' advanced LiDAR-based 360° perception system tailored for inner-city automated driving [13, 17]. The system employs a hierarchical design capable of accurately representing both stationary and mobile elements in a 3D space. Through an optimized point cloud representation, the study demonstrates the system's real-time execution on hardware mounted on vehicles, showcasing its suitability for closed-loop driving functions. This advancement contributes to the overall robustness and effectiveness of LiDAR-based perception systems, addressing key challenges in autonomous navigation and paving the way for enhanced safety and performance.

2.2 Pole Localization

The extraction and localization of poles in LiDAR-based autonomous systems hold significant importance in enhancing the overall performance and accuracy of self-driving vehicles. In the context of autonomous navigation, precise vehicle localization is paramount for optimizing trajectory planning and ensuring accurate positioning.

Lee et al. [11] proposed utilizing pole maps for enhanced vehicle localization, essentially addressing the challenge of trajectory offset corrections. By establishing a pole feature map with detailed spatial information, their method leverages 3D LiDAR for environmental scanning and dynamic thresholding. This enables the extraction of pole-shaped characteristic point clouds along the road, overcoming identification errors common in 3D LiDAR due to scattering effects.

The resulting pole map, utilizing occupancy grid mapping to filter out noise, becomes a valuable reference for localization. As the vehicle moves, continuous detection of pole features enables real-time comparison with the map. When detected features align with the map, the current localized information is updated, contributing to the correction of trajectory information and, consequently, enhancing vehicle localization accuracy. This integration of pole extraction and localization not only improves the precision of the vehicle's position but also aids in the establishment of a high-precision map, crucial for the overall robustness and reliability of LiDAR-based autonomous systems.

2.3 Integration of ICP with Localization

The integration of Iterative Closest Point (ICP) in pole localization methods addresses a critical challenge in autonomous localization within complex urban environments. Other localization methods often struggle with precision degradation due to occlusion caused by dynamic objects like vehicles and pedestrians. Li et al. [12] propose a pole-like feature-based localization framework that leverages the robustness of pole-like objects to overcome occlusion challenges. These features, being taller than objects on the road, remain more consistently visible. Their method, similar to ORIPE, employs a robust clustering algorithm to extract pole-like features from point cloud data, and these features are stitched together across different frames to generate a feature map.

For online localization, a Monte Carlo Localization (MCL) framework combines vehicle motion data with the map-matching result. Notably, an enhanced version of the ICP algorithm, tailored for pole-like feature association, plays a pivotal role in map matching by aligning the state of every particle. This integration ensures precise localization even in the face of challenges like local minima or the robot kidnapping problem [8]. Importantly, this approach claimed to outperform some state-of-the-art localization methods at the time in navigating the intricate and dynamic settings of complex urban environments. The incorporation of ICP, specifically adapted for pole-like features, enhances the reliability and accuracy of the localization framework, making it well-suited for real-world applications in autonomous systems.

3 METHODOLOGY

3.1 Overview

To realize this project, the **O**nline **R**ange **I**mage-based **P**ole **E**xtractor (ORIPE) for long-term localization of autonomous

mobile systems is heavily adopted from [6, 7][1] with minor modifications. ORIPE relies on the range images for pole extraction rather than directly using raw point clouds from LiDAR sensors. Because of their light-weight, natural representation, pole extraction scans are much faster and efficient to operate with. In the mapping phase, the raw point clouds from LiDAR sensors are projected into a range image. Poles are extracted from this range image at their precise positions and the ground-truth poses of the autonomous vehicle are reprojected into the global coordinate system to construct a global map. At the localization phase, the Monte Carlo Localization (MCL) [5] is used to update the relevant weights of the particles matching the pole detections from online sensor data with the poles of the global map. To further improve the matching via pole extraction and MCL localization, Iterative Closest Point (ICP) [18] is introduced to improve alignment of the resulting local maps and the generated global map.

3.2 Range Image Generation

ORIPE revolves around the utilization of range images derived from LiDAR scans to facilitate pole extraction. Building upon prior research [3, 4], it employs a spherical projection for the generation of range images. In this process, every LiDAR point $p = (x, y, z)$ undergoes mapping to spherical coordinates through a function $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, and subsequently to image coordinates given by:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \Phi(x, y, z) = \begin{bmatrix} \frac{1}{2}\left[1 - \arctan(y, x)\pi^{-1}\right] w \\ \left[1 - (\arcsin(zr^{-1}) + f_{\text{up}})f_{\text{down}}^{-1}\right] h \end{bmatrix}$$

Here, (u, v) denotes image coordinates, (h, w) represent the height and width of the desired range image, $f = f_{\text{up}} + f_{\text{down}}^{-1}$ is the vertical field-of-view of the sensor, and $r = ||p_i||_2$ is the range of each point. This mapping yields a collection of (u, v) tuples, each providing image coordinates for a corresponding p_i. Subsequently, these indices facilitate the extraction and storage of p_i's range, as well as its x, y, and z coordinates within the image representation.

3.3 Pole Extraction

Poles are extracted based on the range images produced in the previous step. ORIPE operates on the premise that pole range values are typically significantly smaller than the background. The initial step involves clustering pixels within the range image into distinct regions based on their range values. ORIPE traverses all pixels in the range image from top to bottom and left to right, placing those with valid range data in an open set O. To eliminate ground points, pixels below a specified height are ignored. An examination of neighbors occurs for each valid pixel p, including those to the left, right, and below. If a neighboring pixel with valid data exists and the range difference between the current pixel and its neighbor is below a threshold ε, the current pixel is added to a cluster set c and removed from O. This process is repeated iteratively with neighbors until no neighbor pixel meets the criteria, resulting in a cluster of pixels. After examining all pixels in O, a set C is generated with several clusters, each representing an object. Clusters with pixel counts below a threshold θ are considered outliers and ignored.

The subsequent step involves extracting poles from these objects using geometric constraints. Considering the aspect ratio of each cluster, clusters with an aspect ratio $h : w < 1$ are discarded since the overall interest is in pole-like objects where the height is generally greater than the width. Another observation involves assessing the distance of a pole from background objects; clusters are discarded if the number of points in the cluster with range values smaller than their neighbor is less than δ times the total number of points in the cluster. To leverage 3D coordinates (x, y, z) of each pixel, $\max(z) - \min(z)$ is calculated for each cluster, treating a cluster as a pole candidate only if $\max(z) - \min(z) > \xi$. Additionally, the focus is on poles with a height exceeding θ, introducing a threshold ϕ for the lowest pole position to filter outliers. For each pole candidate, a circle is fitted using the x and y coordinates of all points in the cluster, extracting the center and radius. Candidates with excessively small or large radii and those connected to other objects are filtered out by examining free space around them.

3.4 Mapping

To construct the 2D global pole map for localization, ORIPE heavily adopts the methodology of [15]. This involves partitioning the ground-truth trajectory into shorter segments of equal length. As seen in [15], the inherent inaccuracy of the provided poses for mapping are not well-suited for aggregation as they will generate a suboptimal, noisy submap. Instead, ORIPE selectively utilizes the middle LiDAR scan from each section to extract pole information. The consolidation of multiple overlapped pole detections is achieved by averaging their centers and radii. Subsequently, a counting model is employed to discern and filter out dynamic objects. Only poles that manifest in multiple instances across consecutive sections are considered genuine and included in the final representation.

3.5 Monte Carlo Localization

Monte Carlo Localization (MCL) is a widely employed technique typically realized through a particle filter [5]. MCL operates as a recursive Bayesian filter, estimating the probability density $p(x_t|z_{1:t}, u_{1:t})$ over a pose x_t given all observations

[1] https://github.com/PRBonn/pole-localization

$z_{1:t}$ and motion controls $u_{1:t}$ up to time t. The posterior is iteratively updated through the following expression:

$$p(x_t|z_{1:t}, u_{1:t}) = \eta\, p(z_t|x_t)$$

$$\int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|z_{1:t-1}, u_{1:t-1})\, dx_{t-1}$$

Here, η is a normalization constant, $p(x_t|u_t, x_{t-1})$ denotes a motion model, $p(z_t|x_t)$ is an observation model, and $p(x_{t-1}|z_{1:t-1}, u_{1:t-1})$ represents the probability distribution for the prior state x_{t-1}.

In this specific context, each particle represents a hypothesis for the 2D pose $x_t = (x, y, \theta)_t$ of the autonomous system at time t. Particle poses are updated during movement using a motion model based on control input u_t or odometry measurements. Regarding the observation model, particle weights are updated by evaluating the difference between expected and actual observations, where the observations pertain to pole positions. Online observed poles are matched with map poles through nearest-neighbor search using a k-d tree. The likelihood of each particle is then approximated using a Gaussian distribution:

$$p(z_t|x_t) \propto \exp\left(-\frac{1}{2\sigma_d^2}\sum_{i=1}^{N} d\|z_i^t - z_i^j\|^2\right)$$

Here, d signifies the Euclidean distance between the online observed pole z_i^t and the matched pole in the map z_i^j, given the position of a particle j. N represents the number of matches in the current scan. Following [9], if the number of effective particles falls below a threshold, a resampling process is initiated, and particles are sampled based on their weights.

3.6 Iterative Closest Point

The Iterative Closest Point algorithm (ICP) serves as a foundational technique for aligning and registering 3D point cloud data. Following [18], when given two sets of points $P = \{p_1, \ldots, p_M\}$ and $Q = \{q_1, \ldots, q_N\}$ in $\mathbb{R}^d$, ICP optimizes a rigid transformation on P to align P with Q. This optimization is formulated as follows:

$$\min_{R,t} \sum_{i=1}^{M} \left[\min_{q\in Q} \|Rp_i + t - q\|_2 + I_{SO(d)}(R)\right]$$

Here, $I_{SO(d)}(R)$ is an indicator function ensuring that R is a rotation matrix. The ICP algorithm resolves this problem iteratively through a correspondence step and an alignment step.

Correspondence Step: For each point $p_i \in P$, find the closest point $\hat{q}_i^{(k)} \in Q$ based on the current transformation $(R^{(k)}, t^{(k)})$.

$$\hat{q}_i^{(k)} = arg\min_{q\in Q} \|R^{(k)}p_i + t^{(k)} - q\|_2$$

Alignment Step: Update the transformation by minimizing the ℓ_2 distance between corresponding points.

$$(R^{(k+1)}, t^{(k+1)}) = arg\min_{R,t} \sum_{i=1}^{M} \|R^{(k)}p_i + t^{(k)} - q_i^{(k)}\|_2 + I_{SO(d)}(R)$$

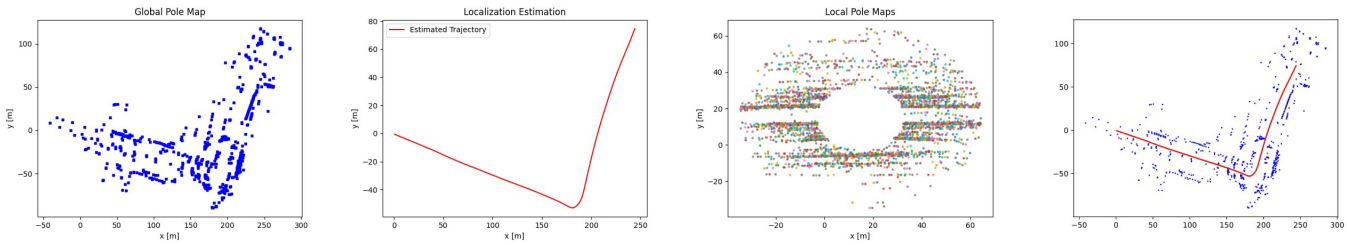
The alignment step is solved in closed form using Singular Value Decomposition (SVD) [16], making ICP an instance of the majorization-minimization (MM) algorithm [10]. The MM algorithm iteratively constructs a surrogate function that bounds the target function from above and minimizes this surrogate function to obtain the next iterate. ICP monotonically decreases the target function until convergence, with the target function for the alignment step serving as a surrogate for the overall target function of the registration problem.

To adapt the ICP formulation for the pole localization problem, it is integrated into the existing pipeline following pole extraction. After ORIPE extracts poles with its range image-based approach, the detected poles are treated as the source point cloud P, and the known locations of poles from the global map as the target point cloud Q. The goal is to align the detected poles with the corresponding poles in the global map. The transformation parameters, such as rotation matrix R and translation vector t, are iteratively updated to minimize the distance between the source and target points. This alignment process enhances the accuracy of local pole maps by rectifying any deviations caused by environmental factors or inaccuracies in the initial pole extraction. ICP provides a robust mechanism to refine the localization of poles in the local map, contributing to an improved match with the global map and enhancing the overall precision of the pole localization system.

4 EXPERIMENTS

4.1 Objective and Hypothesis

The primary objective of these experiments is to establish the baseline performance of ORIPE in regards to the SemanticKITTI dataset. Recognizing that pole extraction and localization are acknowledged enhancers of LiDAR-based system performance—particularly due to their efficiency, lightweight execution, and independence of GPU resources—lays the foundation for our approach. Leveraging the well-established Iterative Closest Point (ICP) algorithm, renowned for its robust alignment capabilities, adds an additional layer to this enhancement strategy. The proposed methodology involves utilizing ICP to align local pole maps, extracted by ORIPE, with the generated global map, with the anticipation that this alignment will contribute significantly to overall performance improvement. The effectiveness of this approach is substantiated through experimentation and visualization, providing



(a) Global Map **(b) Local Estimation** **(c) Local Map Points** **(d) Trajectory Estimation**

Figure 1: ORIPE generated global map, local estimation map, localization points, and final trajectory estimation.

empirical evidence of its impact on LiDAR-based system performance.

4.2 Experimental Setup

To execute these experiments, a VMware Virtual Machine was employed running the Ubuntu 64-bit operating system. The virtual machine was configured with substantial resources, including 8GB of RAM, 35GB of Hard Disk (SCSI) space, and 4 processors, ensuring optimal performance during the experimentation process. The selected distribution was Ubuntu 20.04, complemented by Python 3.8 to facilitate seamless integration with the experimental setup. Visualizations for the ORIPE+ICP framework were generated using custom Python scripts, underscoring the adaptability and versatility of the experimental environment.

4.3 Metrics

In LiDAR localization-based problems, metrics such as precision, recall, and F1-score are commonly employed to quantify the accuracy and performance of localization algorithms. ORIPE natively utilizes a set of predefined metrics, but for the purpose of this project, only the previous three are compared between ORIPE and ORIPE+ICP.

Precision: Indicates the accuracy of positive predictions, i.e., the ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1}$$

Recall: Measures the ability to capture all positive instances, i.e., the ratio of correctly predicted positive observations to the actual positives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2}$$

F1-Score: Balances precision and recall, providing a single metric that considers both false positives and false negatives.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

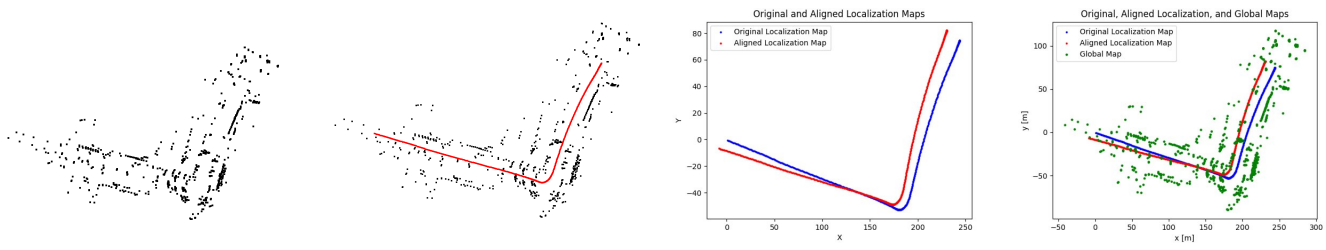
Together, these metrics collectively offer insights into the accuracy, completeness, and overall performance of LiDAR-based localization algorithms.

4.4 Results

The outcomes of our experiments are meticulously presented, combining both quantitative metrics and visualizations to underscore the efficacy of the Iterative Closest Point (ICP) algorithm in enhancing the alignment between local pole maps and the global map produced by the ORIPE algorithm. The intentionally sparse visual representation of the maps, a result of ICP downsampling for matching purposes, offers a clear depiction of the algorithm's impact. Figure 1 provides a snapshot of ORIPE in isolation before the incorporation of ICP, showcasing its commendable performance on the SemanticKITTI dataset. The local estimations and final trajectory align, visually, well with the global map, setting a strong baseline. Table 1 supplements these visual insights with specific ORIPE measurements, offering an intuitive understanding of its performance. Introducing ICP, as depicted in Figure 2, provides an additional perspective, emphasizing ICP's role in realigning the ORIPE local map to the global map with a specified threshold of 0.5. Although an exhaustive exploration of threshold values was not pursued, Table 2 unequivocally demonstrates the substantial performance improvement conferred by ICP upon ORIPE when compared to its original design.

5 ANALYSIS

The integration of the Iterative Closest Point (ICP) algorithm into the ORIPE pole localization method yields several noteworthy outcomes. The alignment and registration capabilities of ICP significantly enhance the accuracy of 3D point cloud data, resulting in a more precise representation of localized poles aligned with features in the global map. This improvement is evident in enhanced precision and recall metrics, indicating a greater likelihood of correctly identifying true positive poles and capturing a higher percentage of actual



(a) Global Map **(b) ICP Matching** **(c) Alignments** **(d) Alignments to Global Map**

Figure 2: Visual difference in alignments of ORIPE and ORIPE+ICP.

Method	Δ_{pos} [m]	$RMSE_{pos}$ [m]	Δ_{lat} [m]	σ_{lat} [m]	Δ_{lon} [m]	σ_{lon} [m]	Δ_{ang} [°]	σ_{ang} [°]	$RMSE_{ang}$ [°]
$ORIPE_{original}$	0.091	0.106	0.066	**0.067**	0.046	0.058	0.084	0.102	0.102
$ORIPE_{mine}$	**0.087**	**0.100**	**0.063**	0.069	**0.044**	**0.055**	**0.071**	**0.094**	**0.095**

Table 1: Original ORIPE measurements [7] compared to the reproduced measurements of this project.

Method	Precision	Recall	F1-Score
ORIPE [7]	0.687	0.439	0.515
ORIPE+ICP	**0.7214**	**0.9581**	**0.8231**

Table 2: Metrics showing improvement with introduction of ICP in ORIPE.

poles present in the environment. The evaluation, encompassing precision, recall, and F1-score, offers a comprehensive and multifaceted assessment of the algorithm's performance. The visual representation of the aligned localization map atop the global map facilitates dynamic observation, providing insights into how well local maps adapt to the global structure. While these results showcase the positive impact of ICP integration, it is crucial to consider computational trade-offs, especially in real-time applications, where increased accuracy needs to be balanced against computational resource requirements.

Despite the improvements achieved through the integration of the Iterative Closest Point (ICP) algorithm into the ORIPE pole localization method, there are several potential limitations to consider. Firstly, the computational demands of ICP, particularly in iterative refinement, might pose challenges for real-time applications, where low-latency responses are crucial. ORIPE runs solely on CPU-based architectures not allowing for faster computations from GPU-based architectures that could also have some affect on ICP's convergence time through its iterative refinement process. Additionally, the effectiveness of ICP is contingent on the quality and accuracy of initial localization estimates generated by ORIPE. In scenarios with poor initial estimates or significant environmental changes, ICP might struggle to converge to accurate solutions. The method's reliance on LiDAR data also means it could be susceptible to adverse weather conditions, such as heavy rain or fog, which may degrade LiDAR sensor performance. Moreover, the proposed evaluation metrics, while providing valuable insights, might not fully capture the system's performance in diverse and complex real-world environments. Addressing these limitations would require a nuanced approach, potentially involving hybrid methods that combine LiDAR data with other sensor modalities for increased robustness and efficiency.

6 DISCUSSION

The identified limitations carry implications for the overall robustness of the system, especially in the context of potential adversarial attacks. One crucial aspect is the computational overhead introduced by ICP, as it may impact the system's responsiveness and real-time capabilities. In scenarios where quick reactions are essential, the extended processing time associated with ICP iterations could be a limiting factor. Moreover, the reliance on accurate initial localization estimates becomes more critical in the face of adversarial attacks, which may deliberately manipulate sensor inputs to mislead the system.

The vulnerability to adverse weather conditions is another factor to consider. Adversarial attacks often exploit unexpected or challenging environmental circumstances to disrupt

sensor-based systems. In the case of heavy rain or fog affecting LiDAR performance, the system's susceptibility to adversarial manipulations may increase. The limitations related to sensor modality and environmental conditions highlight the need for a more comprehensive and adaptable sensor fusion strategy, incorporating data from multiple sources to enhance robustness.

Furthermore, the proposed evaluation metrics, while valuable for assessing system performance, may not fully capture the intricacies of adversarial scenarios. Adversarial attacks often involve subtle perturbations designed to deceive perception algorithms, and the evaluation metrics may need refinement to account for such adversarial-induced anomalies [2]. Developing strategies that consider the specific challenges introduced by adversarial attacks, such as outlier detection techniques and more sophisticated fusion approaches, could bolster the system's resilience and overall performance in adversarial settings.

7 CONCLUSION

The integration of the Iterative Closest Point (ICP) algorithm into the ORIPE pole localization method has demonstrated notable improvements in aligning and registering 3D point cloud data, thereby enhancing the accuracy and fidelity of the localization maps. The evaluation metrics, including precision, recall, and F1-score, collectively indicate a positive impact on the system's overall performance. However, it is crucial to acknowledge the identified limitations, such as computational overhead, sensitivity to adverse weather conditions, and the need for robust initial localization estimates. These limitations highlight the challenges that persist in ensuring the resilience of autonomous systems, especially in the face of potential adversarial attacks. While this work represents a significant stride towards securing autonomous systems, ongoing research and development are imperative to address these limitations comprehensively and fortify the system against adversarial threats.

REFERENCES

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9297–9307.

[2] Sri Hrushikesh Varma Bhupathiraju, Jennifer Sheldon, Luke A Bauer, Vincent Bindschaedler, Takeshi Sugawara, and Sara Rampazzi. 2023. EMI-LiDAR: Uncovering Vulnerabilities of LiDAR Sensors in Autonomous Driving Setting Using Electromagnetic Interference. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 329–340.

[3] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. 2019. Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4530–4537.

[4] Xieyuanli Chen, Ignacio Vizzo, Thomas Läbe, Jens Behley, and Cyrill Stachniss. 2021. Range image-based LiDAR localization for autonomous vehicles. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5802–5808.

[5] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. 1999. Monte carlo localization for mobile robots. In *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, Vol. 2. IEEE, 1322–1328.

[6] H. Dong, X. Chen, S. Särkkä, and C. Stachniss. 2023. Online pole segmentation on range images for long-term LiDAR localization in urban environments. *Robotics and Autonomous Systems* 159 (2023), 104283. https://doi.org/10.1016/j.robot.2022.104283

[7] H. Dong, X. Chen, and C. Stachniss. 2021. Online Range Image-based Pole Extractor for Long-term LiDAR Localization in Urban Environments. In *Proceedings of the European Conference on Mobile Robots (ECMR)*.

[8] Sean P Engelson and Drew V McDermott. 1992. Error correction in mobile robot map learning. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 2555–2556.

[9] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. 2007. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics* 23, 1 (2007), 34–46.

[10] Kenneth Lange and Kenneth Lange. 2013. The MM algorithm. *Optimization* (2013), 185–219.

[11] Sheng-Wei Lee, Peng-Wei Lin, Yuan-Ting Fu, Chih-Ming Hsu, Chen-Yu Chan, Jhih-Hong Lin, and Yen-Hung Chiang. 2020. Improving vehicle localization using pole-like landmarks extracted from 3-D lidar scans. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2052–2057.

[12] Liang Li, Ming Yang, Lihong Weng, and Chunxiang Wang. 2021. Robust localization for intelligent vehicles based on pole-like features using the point cloud. *IEEE Transactions on Automation Science and Engineering* 19, 2 (2021), 1095–1108.

[13] Tobias Nothdurft, Peter Hecker, Sebastian Ohl, Falko Saust, Markus Maurer, Andreas Reschka, and Jürgen Rüdiger Böhmer. 2011. Stadtpilot: First fully autonomous test drives in urban traffic. In *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 919–924.

[14] Jens Rieken and Markus Maurer. 2020. A LiDAR-based real-time capable 3D perception system for automated driving in urban domains. *arXiv preprint arXiv:2005.03404* (2020).

[15] Alexander Schaefer, Daniel Büscher, Johan Vertens, Lukas Luft, and Wolfram Burgard. 2019. Long-term urban vehicle localization using pole landmarks extracted from 3-D lidar scans. In *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 1–7.

[16] Olga Sorkine-Hornung and Michael Rabinovich. 2017. Least-squares rigid motion using svd. *Computing* 1, 1 (2017), 1–5.

[17] Jörn Marten Wille, Falko Saust, and Markus Maurer. 2010. Stadtpilot: Driving autonomously on Braunschweig's inner ring road. In *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 506–511.

[18] Juyong Zhang, Yuxin Yao, and Bailin Deng. 2021. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 7 (2021), 3450–3466.