LiDAR based Landmark and Odometry fusion for precise rail localization without additional infrastructure

Johannes Wortmann, Florian Wulff
Fraunhofer Institute for Open Communication Systems (FOKUS)
Berlin, Germany
johannes.wortmann@fokus.fraunhofer.de
florian.wulff@web.de

Bernd Schäufele, Ilja Radusch
Daimler Center for Automotive Information Technology Innovations (DCAITI)
Berlin, Germany
{bernd.schaeufele; ilja.radusch}@dcaiti.com

***Abstract*—This paper addresses the need for precise train localization within rail networks. Traditional methods face challenges such as high costs, imprecision, and vulnerabilities to disruptions. To overcome these issues, we propose a localization approach without additional infrastructure, i.e., using only existing infrastructure landmarks, specifically catenary poles, using a Deep Neural Network (DNN) within the 3D point cloud captured by a LiDAR sensor. By matching these landmarks to a precomputed two-dimensional HD Map and fusing this information with a state-of-the art LiDAR odometry algorithm, our proof of concept demonstrates the ability to achieve submeter precision for electrified rail segments.**

***Index Terms*—LiDAR, rail, localization, landmark, odometry**

I. INTRODUCTION

Precise localization of trains plays a crucial role in any rail network. It is one of the key factors of ensuring train safety and has a direct impact on network efficiency. The more precise the localization is, the smaller the safety distances between trains in the same network can be selected [1]. This minimizes delays and maximizes network throughput. In addition, precise localization is the basis for various services, such as predictive maintenance of infrastructure or passenger information systems.

Traditionally, localization relies on the installation of extensive dedicated infrastructure, which is both time-consuming and costly to construct and maintain. Furthermore, much of this infrastructure, such as balises and cable-loops, only provides location data when a train is in close proximity. The European Train Control System (ETCS) necessitates continuous train localization to transition from fixed-size safety blocks to moving blocks (ETCS Level 3). For ETCS Level 3, as well as for lower levels like Level 2 and Hybrid Level 3 (HL3), trains must regularly transmit position reports to a trackside Radio Block Centre (RBC) via GSM-R (GSM for Railways), regardless of proximity to installed localization infrastructure [2]. To facilitate this, relative localization between trackside infrastructure points is typically achieved using odometers [3]. Looking ahead, GNSS (Global Navigation Satellite System) is proposed for localization [3], alongside 5G technology (FRMCS) for on-track communication.

Unfortunately, odometers in trains are notoriously imprecise because of the hunting oscillation on railway wheelsets and the reduced friction between wheel and tracks. GNSS on the other hand suffers in specific environments such as tunnels, bridges, or urban canyons. Also, the increasing incidents of attacks on GNSS [4] can compromise localization and operational safety. Moreover, without additional processing, GNSS does not allow track-precise positioning. The rail domain is characterized by stringent regulations and exceptionally high safety and security requirements. As a result, the need for redundant systems or systems that can compensate for each other's shortcomings becomes crucial. Redundant systems ensure that if one component fails or is disrupted, alternative methods for localization and communication can maintain operational integrity. By integrating diverse technologies and approaches, the rail industry can enhance resilience against potential threats and ensure continuous, reliable service while safeguarding the safety of passengers and cargo.

This work seeks to enhance train localization by utilizing already existing infrastructure, not originally intended for localization, as landmarks. Examples of this type of infrastructure include catenary poles, signal posts, lamps, and pillars. Only relying on this type of existing infrastructure drastically reduces both deployment cost and time. In contrast to other environmental features such as vegetation or residential buildings, this infrastructure does not change much with time and is usually owned by the operator of the rail network. The detections of landmarks are mapped to an HD Map for localization, fused with LiDAR odometry.

Of course, detecting these landmarks requires trains to be equipped with suitable sensors. We focus on utilizing LiDAR (Light Detection and Ranging) sensors due to their versatility, allowing them to be employed for various tasks. They are also less impacted by environmental conditions compared to other sensors such as cameras. This is why LiDAR is already heavily used in the automotive domain and is likely to meet the same fate in the rail domain soon. Some industrial rail products already rely on LiDAR, such as the obstacle detection system [5] and the autonomous tram [6] by Siemens Mobility. In [7] the authors show how to classify typical objects in the rail domain. Hence, if LiDAR is installed on the train, it makes sense to leverage the data that is already present to improve the localization as well.

The structure of the paper is organized as follows: Section II provides an overview of existing localization techniques within the rail domain and related areas. This is followed by a description of our developed system in Section III. In Section IV, we present our evaluation methodology along with the results obtained. Finally, Section V concludes with a summary and perspectives for future research.

II. RELATED WORK

A lot of research has been performed in the field of Simultaneous localization and mapping (SLAM). Traditionally, this work aims on ground (UGV) or aerial robots (UAV), structure from motion applications or the automotive domain. A classic example for LiDAR based SLAM is LOAM [8] and its extensions that enhance LOAM by global optimization and loop closure [9]. More recent LiDAR based odometry algorithms include KISS-ICP [10] and CT-ICP [11] or algorithms like Suma++ [12] that leverage semantic information.

Unfortunately, the rail domain differs in some key aspects, which makes applying existing solutions challenging. First, trains usually operate very long routes with higher velocities than robots or cars in an urban setting. Since the precision of most LiDAR or visual odometry approaches is subject to noise, small errors in position estimation, will accumulate dramatically over time (drift). To counter this drift, most of the work relies on loop-closure [13] [14] methods, whose aim is to detect places the vehicle has been to before and correct the map constructed so far and thereby the estimated trajectory. In the rail domain however, this technique is not applicable, because the time between returning to a previous location is far too long. Map matching in the rail domain, generally suffers from very repetitive surroundings and routes with considerably less turns compared to a route in an urban environment. Finally, different error types have different tolerances in the rail domain. While some error along the track (longitudinal error) is rather tolerable (it mostly affects the size of the virtual safety blocks, slightly degrading the networks efficiency), the cross-track error is critical to ensure both safety and efficiency of the network. The obvious exception to this is a single-lane track. Moreover, depending on the grade of automation (GoA), automatic (ATO) or remote train operation (RTO), require precise longitudinal accuracy for reliable on/off-track detection of objects when referencing the train into a virtual sensory environment model.

Previous work in the rail domain tried different approaches to tackle these domain-specific challenges. The authors of [15] created dedicated trackside kilometer posts detected by camera and combined those with LiDAR data for motion and distance estimation. Some research approaches limit their work to specific use cases where other methods, like GNNS, fail. One example for this, is localization inside tunnels [16] using LiDAR and IMU (Inertial Measurement Unit) data or using the disturbance of Earth's magnetic field [17].

In the automotive domain, high-definition (HD) map data is a key feature to enable highly automated driving [18]. Because of its even more constrained nature, we propose a corresponding approach for the rail domain. The authors of [19] implement map matching without LiDAR. Instead, they use a combination of a gyroscope and odometer taking advantage of the inherited "one-dimensional" (1-D) train track profile. The authors of [20] treat the IMU and wheel encoders as primary source and combine it with data from LiDAR, camera, GNSS and map information. In our work, however, we aim to achieve similar localization accuracy by only leveraging LiDAR sensors and map data with excessive landmark matching. The authors of [20] also identify catenary poles using the approach from [21], which is based on range images and originated from the automotive domain. Instead of range images, we use a deep neural network (DNN) to infer pole landmarks directly from the point cloud and fuse them with LiDAR odometry using a HD Map.

III. IMPLEMENTATION

We implemented our system using the Robot Operating System 2 (ROS2) [22] framework. As shown in Figure 1 the proposed framework consists of an offline and a runtime part. The offline components are responsible for creating an HD Map, which combines the information of all rail tracks in the region of the train journey with a map of existing landmarks. In principle, this could be achieved completely manually, but would require tremendous manual effort. Instead, we use a semi-automated process to produce the HD Map. This map is later used at runtime by our probabilistic fusion to match position estimates by the LiDAR odometry algorithm and detected landmarks to actual ones, facilitating train localization. Hence, at runtime we only need two sources of information: the current LiDAR point cloud and our HD Map. The following sections will explain the relevant components in more detail.

A. HD Map

The owner of the German rail track infrastructure, DB InfraGO AG, is publicly providing map information of their rail infrastructure as part of their open data initiative [23]. Although publishing this data is admirable, the data itself is not sufficient to be used as basis for a HD map. This is mainly due to the fact, that the precision of the tracks is about +-5m [23] and multi-track parts of the rail networks are modeled as a single center line with an additional attribute marking the part as multi-track. Of course, one could try to approximate a more detailed track map from duplicating and shifting the center line, but there is a better suited alternative.

For this work, we use OpenStreeMap [24] as foundation for our HD Map. To the best of our knowledge, this is the most accurate publicly available map data about German rail tracks, especially because of the dedicated OpenRailwayMap community [25]. Although quite accurate and much more detailed than the information provided by DB InfraGO AG,

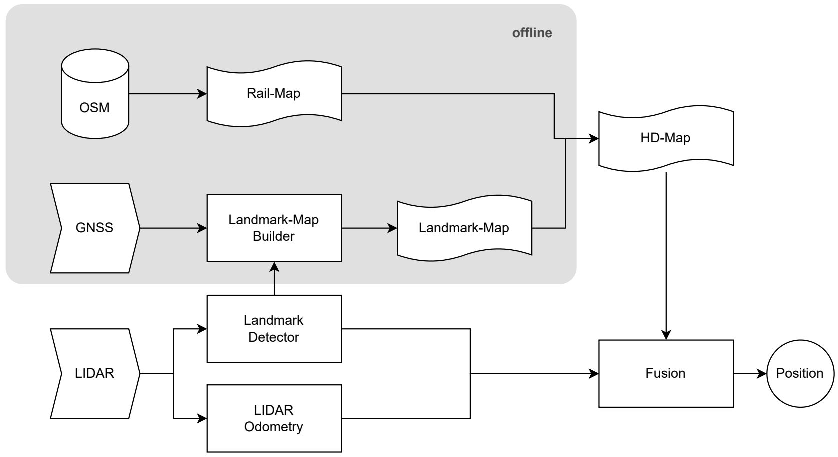

Fig. 1. Overview of the developed system. Rectangles correspond to ROS2-Nodes. The offline components aide in our semi-automated process to create the HD Map, which is used at runtime to localize the train, by matching detected landmarks and odometry estimates.

this data still suffers from some inaccuracies, probably because of the result of manual labor. For example, when compared to satellite images, the track does not maintain a consistent position relative to the two rail heads. Instead, it meanders irregularly in between them. Because of the manual work of the community, the map data also includes small errors from the creators, and each curve and switch are approximated by a different number of line segments.

In an ideal world, all of these issues would be fixed to get a high-quality HD Map, but as shown in Section III-D, if we make our localization framework robust enough, this issue may be negligible.

In section IV we show how we approximated ground truth information from GNSS measurements. This approximation is combined with the clustered results from our landmark detector (subsection III-B) and supported by satellite images. Thereby, it can be used to manually enhance the rail track map by incorporating the landmark locations that constitute our two-dimensional HD map. This semi-automated workflow could be optimized further in the future, if necessary. This is primarily because it works effectively as is, and secondly, the operators of rail networks may already possess this information without making it publicly accessible.

B. Landmark detector

As previously described, we choose pole-like features as our landmarks. In the rail domain these are mostly catenary poles combined with less dominant poles like signal posts, lamps and pillars of bridges etc. Although possible to build a pole detector with traditional methods, we eventually decided to implement a DNN based landmark detector. There are two reasons for this. First, our DNN based approach works more reliable than our previous classical methods, while still offering reasonable runtime performance. Secondly, by using 3D semantic segmentation as our basis, this approach is easier to extend to more complex features than just poles.

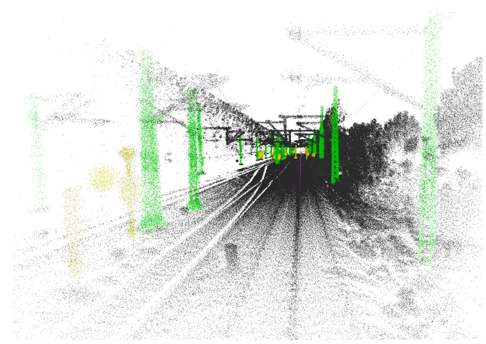

Fig. 2. Segmented pole-features (green) and background-class (black)

To detect our landmarks in the LiDAR point cloud, 2DPASS [26] is trained for the binary segmentation of a background and a catenary pole class (Figure 2). We semi-automatically labeled about 25,000 LiDAR scans. Because points which belong to a pole are much rarer compared to the labels of the remaining points, we use a randomly sampled and equally weighted background class to address this class imbalance during training. As the train is moving during the point cloud acquisition, we apply motion correction on the data [7].

In order to make our approach more robust against outliers, reduce computation time during matching and allow easier evaluation of the HD Map, we work on object-instances rather than the segmented points directly. Because of this, we cluster the segmented pole-points into object instances. Afterwards we perform outlier removal to remove unlikely pole instances, based on number of points, dimensions or orientation. Finally, we make the pole landmarks height-invariant by projecting them to 2D (ground plane). This is particularly important, because the height of the extracted pole instance is influenced by the FOV of the LiDAR, the distance between sensor and pole, and by potential occlusions. Of course, we lose information by doing so, but trade this against greater robustness, when matching those features with the HD Map.

C. LiDAR Odometry

As shown in Figure 1 our developed framework is generic with respect to the used LiDAR odometry algorithm. The only requirement is a ROS2 integration and no strong dependency on any loop-closure techniques. The main purpose of the LiDAR odometry component is to generate motion estimates in between landmarks and estimate the direction of travel.

In our work, we evaluated different approaches, but the selection of the algorithm, once it was tuned for the use case, did not play a big role since they show comparable performance and are all similarly affected by the drift and the inability to perform loop-closure. Since we limit ourselves in this work to a LiDAR-only approach, IMU aided algorithms are excluded, but could be used as well. In our final setup we rely on KISS-ICP to fulfill the task of the LiDAR odometry provider, because of its robustness and accuracy [10], while also easy to integrate.

D. Fusion

Our proof of concept uses a Monte Carlo localization approach to fuse landmark information with the LiDAR odometry estimation.

The state space is parameterized as follows:

$$\begin{pmatrix} e_i & \alpha_i & \theta \end{pmatrix}^T \tag{1}$$

where e_i is the segment id of the current edge in the rail graph, $\alpha_i \in [0, 1]$ is the fraction defining the actual position on the edge e_i and $\theta \in [0, 2\pi]$ being the direction of travel.

This parametrization is rather coarse and intended to show the robustness of the overall approach despite the high state abstraction. In future work, we will investigate the impact of a more detailed modeling of the location state, incorporating the rotational offset between the sensor rig on the locomotive and the wheel axle, which is relevant during turns. Additionally, a lateral offset is beneficial for addressing map inaccuracies and the hunting oscillation on railway wheelsets.

The motion model is defined by the current relative motion estimate of the LiDAR odometry component $v_t = (\hat{x}, \hat{y}, \hat{z})$. Gaussian noise is added with respect to the magnitude of the motion vector, giving $\tilde{v_t}$. This assumes larger measurement errors for larger translations, which is a reasonable assumption for most ICP methods.

The measurement model is used to assess the alignment of a position hypothesis with our HD Map. Here we weight the hypothesis based on two criteria:

- Alignment of the estimated motion with the track map
- Alignment between detected landmarks and the landmarks from the map

To compute the motion term, we calculate two new locations. First, our new location is constrained to the rail graph. For this we find the closest point in the rail map, which is reachable from our previous location on the graph with a travel distance of $\|\tilde{v_t}\| + \epsilon$. Then we calculate the unconstrained new position, by simply applying our motion estimate $\tilde{v_t}$ to the previous location. The difference of the two location estimates constitutes the motion term of the measurement model, which then gets weighted by a robust weighting function to give the motion weight w_{motion}

To compute the landmark matching term, we first need to match the individual landmarks in each detection frame with the landmarks from the map. We transform every landmark detection in the current LiDAR frame to the local frame of the hypothesis and match it to the closest (Euclidean) landmark in the map with an additional rejection threshold τ_{land} to exclude outliers. This means we weight only based on the detected landmarks in the current LiDAR frame and ignore missed landmarks from the map. This is a simple robustification in case of occlusions (e.g., other trains on parallel tracks). In the future, this could be further enhanced by implementing a ray-tracing approach to check if the landmark from the map could have been observed in the point cloud. Note that this matching is non-exclusive allowing multiple detected landmarks to be matched against the same landmark from the map. Given the landmark matching, we compute the landmark weight w_{land} using a robust distance metric

$$\delta_i = min\left(\|\vec{x}_{i,map} - \vec{x}_{i,detect}\|, \tau_{land}\right) \tag{2}$$

and gaussian weighting

$$w_{land} = \lambda_l \sum_i \exp -\frac{\delta_i^2}{2\sigma^2} \tag{3}$$

, where i is the index for all matched landmark detections and λ_l a normalization factor. The full measurement model then combines the two weights w_{land} and w_{motion} to form a single weight for each hypothesis.

IV. EVALUATION

To evaluate performance, we applied our method to 11 recorded trips within the rail network of Berlin and its greater surroundings. The trips vary in length, ranging from $0.4\,\mathrm{km}$ to $6\,\mathrm{km}$ and are set in diverse environments, including both urban and rural areas. Furthermore, the abundance of poles differs dramatically between the trips. Fully electrified rail lines feature a high density of poles, whereas partially electrified tracks have fewer poles and only sporadic occurrences of other pole-like structures, such as signals.

Trips 0 to 6 took place on electrified multi-track segments (at least two parallel rail tracks) of the network in an urban setting, while trip 8 to 10 were mostly unelectrified single-track segments in more rural environments. Trip 7 is the transition between the two groups of trips but being more similar to the second group than the first. The total evaluated trip length amounts to $38.5\,\mathrm{km}$, including multiple train stations with several platforms and parts with up to 11 parallel tracks. Since our approach in its current form requires an initial location, we provide the very first GPS message to initialize our algorithm.

A. GNNS based approximation of ground truth

The downside of the data set used is its lack of a reliable ground truth. To overcome this issue and hence be able to assess the quality of our method, we approximate the actual ground truth as closely as possible with the available data.

The best source available in our recorded trips is based on a Dual-Antenna GNSS sensor with integrated IMU and RTK correction, which, after manual inspection, produces position estimates with sufficient precision most of the time. Unfortunately, the GNSS readings are partly noisy or completely missing, due to bridges, tunnels, or other disturbances, as shown in Figure 3. Thus, we need a reliable way to correct errors and interpolate missing sensor messages. In addition, we need to interpolate regular GNSS messages, as the sensor frequency of the GNNS sensor is lower than that of our LiDAR. This allows for a more precise error calculation at each position update of our method (one per LiDAR measurement).

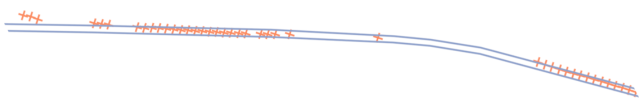

Fig. 3. Missing or inaccurate GNSS messages (orange) along rail track (blue) in an urban environment including a train station and multiple bridges

To achieve this, we used the following semi-automated approach: First, we match the initial GNSS location L_0 to the closest edge on the rail graph and manually verify that it was not matched to an incorrect parallel rail track. This matched initial location l_0, is now the initial location of our approximated GT.

Then, for every following GNNS location $L_i, i > 0$ do

1) Find a GNNS location L_{i+k} in the future ($k > 0$), which is at least τ_{GT} meters away from L_i and match it to the closest edge of the rail graph, giving $\hat{l}_{i+k}$.
2) Calculate the shortest possible path in the rail graph from the previous determined GT point on the graph l_{i-1} to the temporary look-ahead $\hat{l}_{i+k}$
3) Match L_i to the closest edge on the shortest path, giving l_i.
4) Interpolate along the matched shortest path at every LiDAR timestamp that fits between the timestamps of L_{i-1} and L_i. This assumes constant velocity between the two GNSS measurements.

Interpolating along the path, is of particular importance for gaps in the GNSS message stream (e.g., because of bridges), where interpolating along the direct connecting line would lead to wrong results. The use of the temporary look-ahead is mostly relevant to avoid precision issues with inaccuracies in very narrow situations (e.g., the beginning of a switch).

It is important to note, that this approach is a heuristic and requires manual verification of the result. Therefore, we compared the approximated GT trajectories to the rail map and further visually verified its accuracy at critical sections (e.g., switches) using images from a camera installed in the train and the projected rail and pole maps into said images.

By using this approach, we can ensure that there is no relevant cross-track (lateral) error in the approximation. However, the same cannot be said for the longitudinal error, which is affected more by the environment. In urban areas, with numerous visual reference points (such as poles, switches, and turns), there are no large errors along the track in our ground truth (GT) approximation. In contrast, rural areas with fewer reference points and longer straight track segments do not allow for the same level of certainty. Nonetheless, these regions typically experience fewer signal disturbances, resulting in a better GNSS signal quality in the first place, which helps mitigate these issues. Unfortunately, the dataset includes only one recording per trip, requiring the use of the same data for both GT approximation and evaluation, which may introduce bias. However, the use of different sensors and manual image verification mitigates this issue to a large extent.

B. Experimental results

We evaluate our approach using two different metrics: the absolute and the relative trajectory error. The absolute trajectory error quantifies the difference in location between the ground truth approximation and the position estimation of our method at the same point in time. In contrast, the relative trajectory error measures the difference in displacement between the previous timestamp t_{i-1} and the current timestamp t_i, comparing the approximated GT and our method.

The results of the comparison between our approach and the approximation based on GNSS are presented in Table I. Since both positions are matched to the same rail map, and we have verified that our estimation remains on the correct track without drifting onto neighboring tracks, all reported errors are confined to the track itself (i.e., longitudinal errors).

Our findings show very small errors when poles are present and larger errors without. This suggests that landmark matching reliably helps mitigate drift. If poles are frequently available our method achieves sub-meter accuracy. Overall, our method produces at least comparable accuracy to GNSS by utilizing only a single forward-facing LiDAR sensor and a map. Notably, the GNSS messages from our trips required corrections to obtain a more accurate ground truth approximation. Therefore, our method is indeed more precise than GNSS in the presence of poles, particularly in environments with signal disturbances such as bridges, train stations, and tunnels.

V. CONCLUSION

In this paper, we presented an approach to robustly localize trains using only a LiDAR sensor and a precomputed two-dimensional HD Map with rail track and landmark locations. All of the landmarks are already existing static parts of the infrastructure of the rail network. Because of this, no dedicated localization infrastructure needs to be installed, making this approach very cost-effective and easy to maintain. In our proof of concept, we use pole-like features (mostly catenary poles) as landmarks and achieve sub-meter precision for electrified parts of the rail network. Since LiDAR sensors are very likely to be installed in future railway vehicles, our approach is a promising option for a cheap and reliable redundant system to GNSS and already installed localization infrastructure.

TABLE I
EXPERIMENTAL RESULTS (ALL ERRORS ALONG THE TRACK)

#Poles	Trip	Length	Absolute Trajectory Error (m) RMSE	MEAN	MEDIAN	STDDEV	Relative Trajectory Error (m) RMSE	MEAN	MEDIAN	STDDEV
Many	0	3.3 km	0.55	0.20	0.12	0.51	0.14	0.04	0.00	0.13
	1	2.5 km	0.37	0.18	0.11	0.33	0.12	0.03	0.00	0.11
	2	3.4 km	0.79	0.48	0.32	0.62	0.15	0.05	0.00	0.14
	3	4.0 km	0.27	0.20	0.15	0.18	0.14	0.05	0.00	0.13
	4	0.4 km	0.08	0.04	0.02	0.07	0.03	0.01	0.00	0.03
	5	0.6 km	0.12	0.06	0.02	0.10	0.06	0.01	0.00	0.06
	6	2.9 km	0.34	0.17	0.10	0.29	0.12	0.03	0.00	0.11
Few	7	5.5 km	2.12	1.50	1.13	1.51	0.44	0.10	0.00	0.43
	8	4.9 km	7.53	5.03	3.92	5.61	0.42	0.10	0.00	0.41
	9	6.0 km	5.21	3.57	2.90	3.80	0.68	0.13	0.00	0.67
	10	5.0 km	10.84	9.30	8.31	5.56	0.53	0.12	0.00	0.52

In future work, we aim to extend the basic formulation of our probabilistic fusion to better address map and sensor inaccuracies, resulting in even greater precision. Also, improving the automatic map generation is a promising way to proceed to further reduce the deployment cost of the developed system.

ACKNOWLEDGMENT

We would like to issue a special thanks to Siemens Mobility GmbH for providing us with valuable rail domain data.

REFERENCES

[1] J. Beugin, C. Legrand, J. Marais, M. Berbineau, and E.-M. El-Koursi, "Safety appraisal of GNSS-based localization systems used in train spacing control," *IEEE Access*, vol. 6, pp. 9898–9916, 2018.

[2] D. Hansen, M. Leuschel, P. Körner, S. Krings, T. Naulin, N. Nayeri, D. Schneider, and F. Skowron, "Validation and real-life demonstration of ETCS hybrid level 3 principles using a formal B model," *International Journal on Software Tools for Technology Transfer*, vol. 22, 2020.

[3] J. Marais, J. Beugin, and M. Berbineau, "A Survey of GNSS-Based Research and Developments for the European Railway Signaling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2602–2618, 2017.

[4] V. Renaudin and F. Le Bourhis, "The Growing Importance of Alternative Indoor and Seamless Positioning and Navigation in Response to Escalating GNSS Attacks," in *2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2024, pp. 1–6.

[5] Siemens Mobility, "Innovative obstacle detection system is being tested for the first time on the Berlin S-Bahn," https://press.siemens.com/global/en/pressrelease/innovative-obstacle-detection-system-being-tested-first-time-berlin-s-bahn, accessed: 2024-09-03.

[6] Siemens Mobility, "Siemens Mobility presents world's first autonomous tram," https://press.siemens.com/global/en/pressrelease/siemens-mobility-presents-worlds-first-autonomous-tram, accessed: 2024-09-03.

[7] F. Wulff, B. Schäufele, J. Pfeifer, and I. Radusch, "Railway LiDAR semantic segmentation based on intelligent semi-automated data annotation," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. IEEE, 2024, pp. 1–7.

[8] J. Zhang, S. Singh *et al.*, "LOAM: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[9] L. Liao, C. Fu, B. Feng, and T. Su, "Optimized SC-F-LOAM: Optimized Fast LiDAR Odometry and Mapping Using Scan Context," in *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, 2022, pp. 1–6.

[10] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[11] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure," 2021.

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

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

[14] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[15] D. Wang, X. Shi, H. Zhang, J. Huo, and C. Cai, "Visual Landmark-Aided LiDAR–Inertial Odometry for Rail Vehicle," *IEEE Sensors Journal*, vol. 24, no. 17, pp. 27 653–27 665, 2024.

[16] X. Dai, W. Song, Y. Wang, Y. Xu, Y. Lou, and W. Tang, "LiDAR–Inertial Integration for Rail Vehicle Localization and Mapping in Tunnels," *IEEE Sensors Journal*, vol. 23, no. 15, pp. 17 426–17 438, 2023.

[17] T. Strang, A. Lehner, O. Heirich, B. Siebler, and S. Sand, "Train-Localization in Tunnels using Magnetic Signatures," in *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2024.

[18] K. Massow, B. Kwella, N. Pfeifer, F. Häusler, J. Pontow, I. Radusch, J. Hipp, F. Dölitzscher, and M. Haueis, "Deriving HD maps for highly automated driving from vehicular probe data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1745–1752.

[19] S. Saab, "A map matching approach for train positioning. I. Development and analysis," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 467–475, 2000.

[20] Y. Wang, W. Song, Y. Zhang, F. Huang, Z. Tu, R. Li, S. Zhang, and Y. Lou, "Four years of multimodal odometry and mapping on the rail vehicles," *Journal of Field Robotics*, vol. 41, no. 2, pp. 227–257, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22256

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

[22] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science robotics*, vol. 7, no. 66, 2022.

[23] DB InfraGO AG, "Streckennetz der DB InfraGO AG," https://www.govdata.de/suche/daten/streckennetz-der-db-netz-ag, accessed: 2024-11-11.

[24] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[25] M. Reichert and OpenStreetMap Foundation, "OpenRailwayMap," https://wiki.openstreetmap.org/wiki/OpenRailwayMap, accessed: 2024-11-07.

[26] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds," in *European Conference on Computer Vision*. Springer, 2022.