Point Cloud Map Generation and Localization for Autonomous Vehicles Using 3D Lidar Scans

Alwin Poulose*
Center for ICT and Automotive Convergence
Kyungpook National University
Daegu, South Korea
alwinpoulosepalatty@knu.ac.kr

Minjin Baek*
Center for ICT and Automotive Convergence
Kyungpook National University
Daegu, South Korea
mbaek@knu.ac.kr

Dong Seog Han†
School of Electronics Engineering
Kyungpook National University
Daegu, South Korea
dshan@knu.ac.kr

***Abstract*—Autonomous vehicles are the future intelligent vehicles, which are expected to reduce the number of human drivers, improve efficiency, avoid collisions, and become the ideal city vehicles of the future. To achieve this goal, vehicle manufacturers have started to work in this field to harness the potential and solve current challenges to achieve the desired results. In this sense, the first challenge is transforming conventional vehicles into autonomous ones that meet users' expectations. The evolution of conventional vehicles into autonomous vehicles includes the adoption and improvement of different technologies and computer algorithms. The essential task affecting the autonomous vehicle's performance is its localization, apart from perception, path planning, and control, and the accuracy and efficiency of localization play a crucial role in autonomous driving. In this paper, we describe the implementation of map-based localization using point cloud matching for autonomous vehicles. The Robot Operating System (ROS) along with Autoware, which is an open-source software platform for autonomous vehicles, are utilized for the implementation of the vehicle localization system presented in this paper. Point cloud maps are generated based on 3D lidar points, and a normal distributions transform (NDT) matching algorithm is used for localizing the test vehicle through matching real-time lidar measurements with the pre-built point cloud maps. The experiment results show that the map-based localization system using 3D lidar scans enables real-time localization performance that is sufficiently accurate and efficient for autonomous driving in a campus environment. The paper comprises the methods used for point cloud map generation and vehicle localization as well as the step-by-step procedure for the implementation with a ROS-based system for the purpose of autonomous driving.**

***Index Terms*—localization, map generation, map matching, point cloud data, Autoware, normal distributions transform (NDT), autonomous driving**

I. Introduction

Vehicle localization is crucial for autonomous driving, making it possible to accurately and efficiently track the vehicle location as well as orientation with respect to the reference frame. Without highly accurate and reliable localization, it is impossible to have autonomous vehicles safely drive themselves on the road with confidence. Localization methods for autonomous vehicles can be categorized into three types: global localization [1], absolute localization [2], and relative localization [3].

*These authors contributed equally to this work.
†Corresponding author: Dong Seog Han.

In global localization, the localization system uses the information from the global navigation satellite system (GNSS), including the latitude, longitude, altitude/elevation, and heading information. The primary sources in the GNSS-based localization system are the Global Positioning System (GPS), BeiDou Navigation Satellite System (BDS), GLONASS, and Galileo. Real-time kinematic positioning (RTK), precise point positioning (PPP), and GNSS/inertial navigation system (INS) are the three primary accurate positioning methods for GNSS-based vehicle localization at present. However, for its optimal localization performance, the GNSS-based localization system requires an unobstructed view of the sky, which is difficult to achieve in urban driving environments.

In the case of absolute localization, the localization system estimates the vehicle's position and orientation relative to the local features or landmarks from the map. The vehicle is localized by estimating the location of the best possible match between online measurements and pre-built maps. Matching algorithms such as the perfect match (PM) [4], normal distributions transform (NDT) [5], and iterative closest point (ICP) [6] are involved in the localization process. Sensors such as lidars and cameras can be utilized for map-based localization approaches including point cloud matching and landmark search [7]. Creating accurate high-resolution maps for this type of localization can be challenging, and the matching error and process time from localization adversely affect the localization performance.

In the case of relative localization, the localization system estimates the vehicle location relative to the arbitrary origin (e.g., the start location). The vehicle location is estimated within the local map that is dynamically built, and the sensor measurements are integrated over time. The most common algorithms for this localization approach are simultaneous localization and mapping (SLAM) [8] and lidar odometry and mapping (LOAM) [9]. For these localization methods, measurements from various sensors such as inertial measurement units (IMUs), wheel encoders, cameras, and lidars can be used. IMUs measure acceleration and yaw rate as well as roll, pitch, and yaw, and based on these measurements it is possible to estimate the relative vehicle position and orientation. Wheel encoders measure individual wheel rotation speed. Camera and lidar sensors provide information about the vehicle's surroundings from which we can compute the relative movement from frame to frame. The fusion localizer uses the results from multiple types of sensors and estimates the vehicle location and orientation. One of the notable challenges of relative localization is drift error accumulation which reduces the localization performance.

Autonomous driving systems generally require a centimeter-level localization accuracy, and the use of lidar for localization is one of the most appropriate solutions considering its capability of capturing 3D information about the road geometry and driving environment with a high precision [10]. The vehicle localization system described in this paper is based on the absolute localization that employs a map-based localization method using 3D lidar scans. The experiment setup includes a lidar sensor that is mounted on top of our test vehicle. We used the Robot Operating System (ROS) [11] along with Autoware open source software [12] for the implementation of the vehicle localization system in a real-time driving environment. ROS is an open-source middleware that is often used in automation and robotics. ROS employs a graph architecture where multiple software modules called nodes exchange messages by publishing and subscribing to software buses called topics. ROS makes it possible to establish connection between various modules developed for autonomous driving. The transform library for ROS, which is known as the tf library, is used to manage coordinate frames and transformations [13]. We also discuss the process of the NDT scan matching method and describe the steps involved in point cloud map generation as well as vehicle localization with Autoware. Finally, we present the map generation results obtained in a university campus environment along with the vehicle localization results acquired in real-time driving conditions.

The rest of the paper is organized as follows. Section II discusses the related work on vehicle localization in autonomous driving systems. In Section III, we describe the vehicle localization architecture along with point cloud map generation and vehicle localization implementations. Section IV discusses the experiment results, and Section V concludes this paper with future research directions.

II. Related Work

Localization and mapping in autonomous driving systems can be achieved in different ways, including absolute positioning sensors [14], odometry/dead reckoning [15], GNSS-IMU fusion [16], SLAM [17], and map-based localization [18]. This paper mainly focuses on map-based localization approaches, which can be divided into landmark search based localization [19] and point cloud matching based localization [20], [21]. Among these approaches, the point cloud matching based method that uses 3D lidar scans is employed for the vehicle localization system described in this paper.

Many localization approaches have been proposed in the literature [7], [22], [23], which can be categorized into sensor-based localization techniques [24], cooperative localization techniques [25], and data-fusion-based techniques [26]. In sensor-based localization techniques, the system uses GNSS/IMU, camera, radar, lidar, and ultrasonic sensors. However, the installation cost and computational complexity of the sensor data make these systems challenging for real-time vehicle localization. In cooperative localization techniques, the system uses wireless communication technologies such as Wi-Fi, cellular, and ultra-wideband (UWB) for the purpose of localization [22]. Cooperative localization techniques can be mainly divided into vehicle-to-vehicle localization (V2V) and vehicle-to-infrastructure localization (V2I). The latency, packet loss, channel congestion, network security, and dependence on connectivity are among the main challenges of the cooperative localization approaches. Data-fusion-based localization techniques include multisensor data fusion and map-based data fusion [23]. Multisensor data fusion can be employed to fuse on-board sensor data with GNSS data to improve the localization performance [27]. In the case of map-based data fusion, vehicle localization is performed based on on-board sensor measurements in conjunction with pre-built digital maps [19]. Data-fusion-based localization can play a significant role in autonomous driving, particularly for its capability to provide a cost-efficient localization solution [23]. However, the GNSS integrity and driving in complex environments present challenges, and this approach still needs improvement to ensure the localization accuracy and robustness in real-time driving environments.

III. Vehicle Localization

Localization in autonomous driving systems provides the vehicle location and orientation with respect to the reference frame. A general architecture that can be used for vehicle localization in autonomous driving systems is shown in Fig. 1. For the purpose of self-localization, autonomous vehicles can collect data from various sources including GNSS receivers, wheel encoders, IMUs, lidars, cameras, and maps. The preprocessing stage involves various types of operations including downsampling, feature extraction, and concatenation. For a localizer, different localization approaches, such as GNSS-based localization, dead reckoning, map-based localization, and fusion-based localization, can be considered. The localization results are then sent to planning and control systems for safe and efficient execution of autonomous driving. Note that the illustration in Fig. 1 is an example of the general architecture for vehicle localization and that the system architecture can be designed differently—for example, with additional types of sensors and fusing data at a different level of abstraction. As an example of this, the results from the preprocessing stage can also be directly sent to the fusion localizer for lower-level fusion, but the representation for such data flows is omitted in Fig. 1 for the sake of simplicity.

In this paper, we employ a map-based vehicle localization approach using 3D lidar scans, which is well capable of

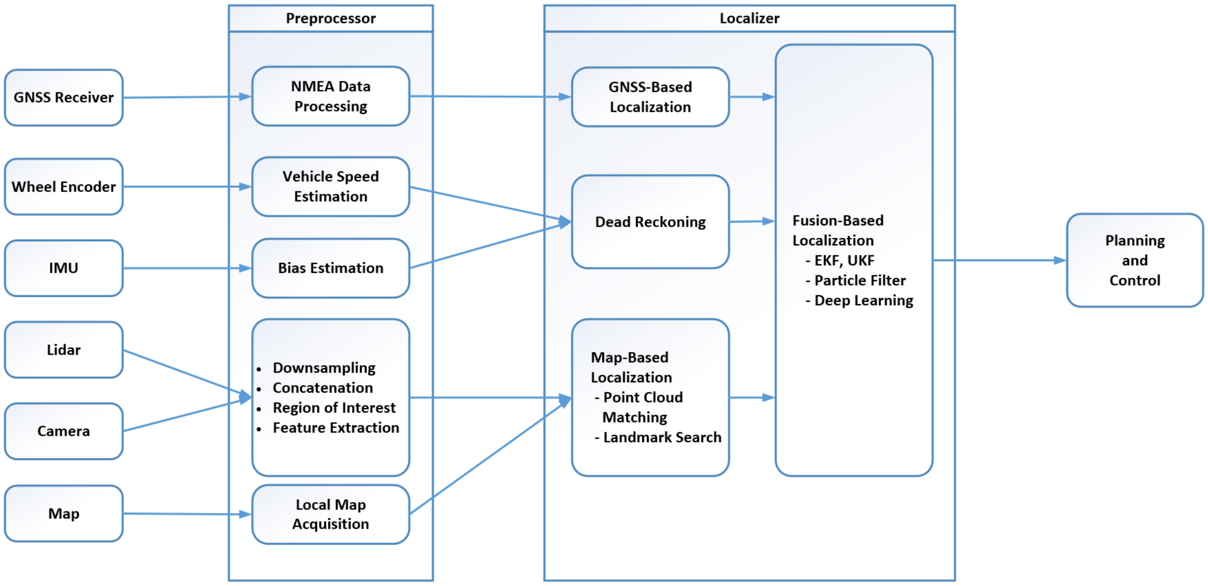


Fig. 1. Localization architecture in autonomous driving systems.

achieving the required localization accuracy and reliability for autonomous driving. For the implementation of vehicle localization described in this work, we used a local machine running on Ubuntu 18.04 LTS, and our development environment setup included ROS Melodic and Autoware 1.14.0. A 128-channel lidar (Ouster OS1-128) was used for data collection for map generation and vehicle localization. Prior to vehicle localization, we first generated point cloud maps based on the lidar data we previously recorded in the form of a rosbag file, which stores ROS message data. In order to perform vehicle localization, we used the NDT matching method for 3D lidar scan matching, which can yield localization accuracy on the order of centimeters [12]. The process of NDT scan matching as well as the steps for performing point cloud map generation and vehicle localization are presented in the following subsections.

A. NDT Scan Matching

The NDT scan matching method [5], [20], [21] attempts to match the lidar points from each received scan to the probability functions obtained from the reference point cloud map. In order to transform the point cloud map into NDT cells, the point cloud map is first discretized into fixed grid cells, and then a probability distribution is assigned to each grid cell by computing the mean and covariance. For vehicle localization, the rotation and translation parameters for matching each lidar scan received are optimized by minimizing the score function with Newton's method. This optimization process is repeated until the convergence criteria are met or the iteration limit is reached. The process of the NDT scan matching method can be summarized as follows:

1) Discretize the reference scan data into grid cells.
2) Calculate the mean and covariance for each grid cell.
3) Initialize the rotation and translation parameters.
4) Based on the parameters, transform the input scan into the coordinate frame of the reference scan.
5) Evaluate the current parameters with the NDT score function.
6) Calculate and update the transformation parameters using Newton's method.
7) Repeat Steps 4–6 until converged or iteration limit is reached.

B. Point Cloud Map Generation

In order to perform map-based vehicle localization using 3D lidar scans, it is necessary to first gather lidar data and generate lidar point cloud maps for the driving test area. This subsection describes the necessary steps for generating point cloud maps with Autoware from lidar data previously recorded in the form of a rosbag file. The lidar data we used in this work were collected with a single lidar sensor. Regarding building the ROS driver for the Ouster lidar and recording and replaying the lidar data in the ROS environment, one may refer to the instructions available at the GitHub page for Ouster SDK[1] . The steps for generating a point cloud map by utilizing the NDT mapping implementation in Autoware can be summarized as follows:

1) Launch Autoware Runtime Manager.
2) Specify the replay and metadata parameters for the lidar to publish ROS topics from recorded data.
3) Load the previously recorded rosbag file.
4) Set `/use_sim_time` to true to work with a simulated clock.
5) Define the transform and load the vehicle model.
6) Activate `ndt_mapping` in the runtime manager.
7) Relay the point cloud topic to `/points_raw`.

[1] https://github.com/ouster-lidar/ouster_example

8) Open RViz for visualization of the mapping process.
9) Start the mapping process in the runtime manager.
10) Save the point cloud map and stop the mapping process when mapping is complete.

For interested readers, we created a detailed step-by-step tutorial video for generating a point cloud map with Autoware and made it available online[2].

C. Map-Based Vehicle Localization Using Lidar Data

The map-based vehicle localization in this work is performed with 3D lidar scan matching. The rotation and translation parameters for matching each lidar scan with the point cloud map are estimated through the NDT scan matching method. This subsection describes the procedure for performing vehicle localization with Autoware, with an assumption that the point cloud map for the area of interest is available for use. The map-based vehicle localization can be performed based on either real-time lidar data or previously collected lidar data recorded in the form of a rosbag file. Although the procedure described below is intended for the latter, one can simply replace the steps related to utilizing a rosbag file with steps relevant to establishing the connection between the lidar sensor and the local machine. The steps for performing vehicle localization by using the NDT scan matching implementation in Autoware can be summarized as follows:

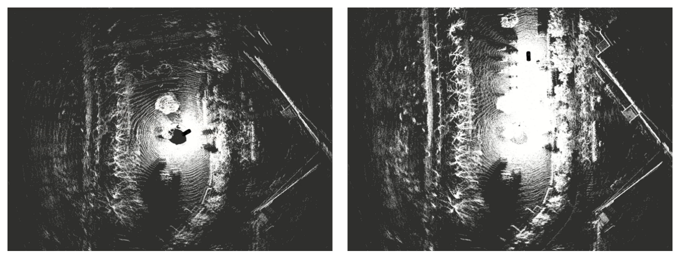
1) Launch Autoware Runtime Manager.
2) Specify the replay and metadata parameters for the lidar to publish ROS topics from recorded data.
3) Load the previously recorded rosbag file.
4) Set `/use_sim_time` to true to work with a simulated clock.
5) Define the transform and load the vehicle model.
6) Load and activate the point cloud map.
7) Activate `voxel_grid_filter` in the runtime manager.
8) Activate `ndt_matching` in the runtime manager.
9) Relay the point cloud topic to `/points_raw`.
10) Open RViz for visualization of the localization process.
11) Provide initial pose information for NDT scan matching.
12) Start the localization process in the runtime manager.

We also created a detailed step-by-step tutorial video for performing vehicle localization with Autoware and provided it online for those who are interested[3].

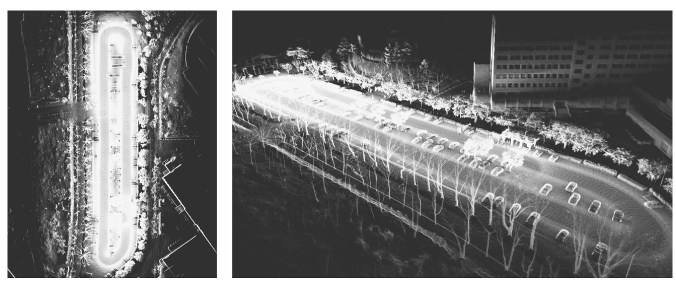
(a)

(b)

Fig. 2. Experiment setup and driving test environment. (a) Test vehicle and sensor configuration; (b) scenes from the driving test area.



(a)



(b)

Fig. 3. Point cloud map of the student parking lot at Kyungpook National University. (a) Map generation process; (b) visualization of the point cloud map.

IV. Experiments and Results

For the implementation and testing of the vehicle localization system described in this paper, our test vehicle was equipped with an Ouster OS1-128 lidar, which we mounted on top of the vehicle roof rack such that it provides a complete view of the vehicle surroundings. The experiment setup also consisted of a local machine equipped with Intel Core i9-9900K CPU, Nvidia Titan RTX, and 32 GB of memory. We used the ROS and Autoware open source software for point cloud map generation and vehicle localization in a real-time driving environment. For an initial performance evaluation of the vehicle localization system, we conducted experiments in the student parking lot at Kyungpook National University, located in Daegu, South Korea. The sensor configuration and the driving test area are shown in Fig. 2.

Prior to vehicle localization experiments we first generated the point cloud map of the driving test area as shown in Fig. 3. Fig. 3a shows the images of the accumulated point cloud taken during the mapping process, and Fig. 3b shows the complete point cloud map for the driving test area. For the purpose of

[2]https://www.youtube.com/watch?v=ZGqG9sYpKns&t=9s
[3]https://www.youtube.com/watch?v=1SFw0eP7Ilc

map generation we recorded lidar data in the form of a rosbag file while driving once around the test area. The vehicle was driven at a speed lower than 2.5 m/s in order to minimize the lidar scan distortion. The point cloud map was created by utilizing the NDT mapping implementation in Autoware, and the parameters for map generation were defined as follows: a resolution of 1 m, a maximum iteration number of 30, a voxel grid size of 1 m, a maximum scan range of 200 m, a minimum scan range of 5 m, and a minimum add scan shift of 1 m.

The results from our vehicle localization experiments conducted in a real-time driving environment are shown in Fig. 4. Both the forward-looking scene and the vehicle localization result are shown here for different time instances, and the location as well as the orientation with respect to the map coordinate frame at each time instance are highlighted with a red circle. For the test run, the vehicle was driven around the test area in a counter-clockwise direction at a speed of about 4 m/s. We used the previously built point cloud map shown in Fig. 3 and the NDT scan matching implementation in Autoware in order to perform the map-based localization using 3D lidar scans. Similar to the parameters set for map generation, the resolution and the maximum iteration number for NDT scan matching were set to 1 m and 30, respectively. The initial pose information for NDT scan matching was inputted at the start of each test run. The experiments for real-time vehicle localization were carried out on a different day than the day we recorded the lidar data for point cloud map generation. This is because the number as well as the location of the vehicles parked in the driving test area change from day to day, which is important for testing the robustness of the map-based localization approach. From this initial performance evaluation, we found that the map-based localization system described in this paper is capable of achieving the level of localization accuracy and reliability required for autonomous driving in a campus environment.

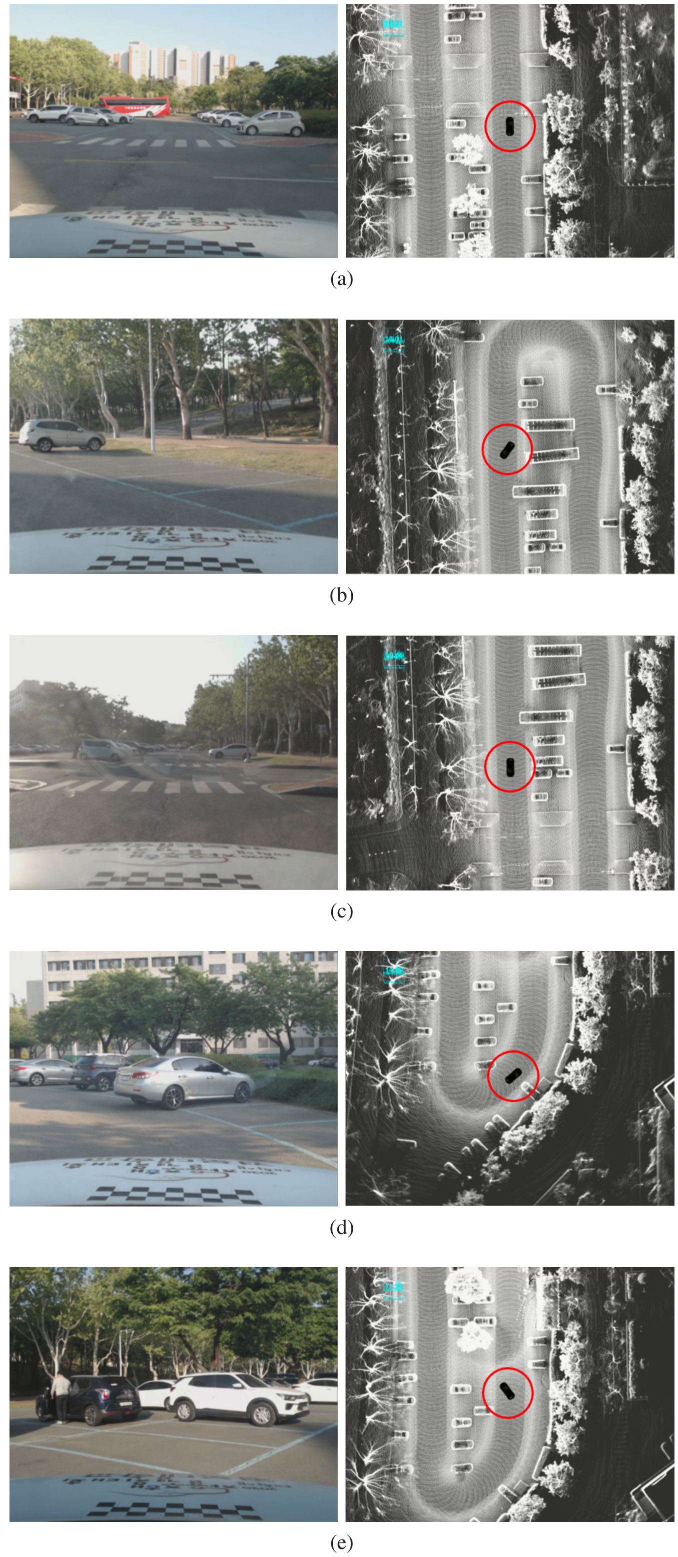


Fig. 4. Qualitative results of the real-time vehicle localization experiment. (a) $t = 26$ s; (b) $t = 53$ s; (c) $t = 59$ s; (d) $t = 100$ s; (e) $t = 113$ s.

V. Conclusion

In this paper, we presented the implementation of the vehicle localization system based on lidar scan matching. Prior to conducting vehicle localization experiments, point cloud maps for the driving test area were created based on the lidar data we previously recorded. We implemented map-based localization by utilizing the NDT scan matching method for matching lidar measurements with the point cloud maps in a real-time driving environment. In addition, we provided practical step-by-step procedures for the implementation of point cloud map generation and vehicle localization in a ROS-based system. Through real-time vehicle localization experiments conducted with our test vehicle, we demonstrated that the map-based localization system described in this work achieves a reasonable accuracy for real-time vehicle localization without significant computational complexity.

The future scope of this work is to improve the real-time localization performance through parameter optimization and also to create highly accurate point cloud maps by utilizing other mapping schemes. NDT scan matching parameters such as the voxel grid size and maximum scan range will be further optimized to enhance the real-time computation performance while preserving the localization accuracy required for autonomous driving. Recently released open-source SLAM implementations that employ a high-frequency IMU and a GNSS receiver along with a 3D lidar sensor will be considered for improving mapping accuracy. In addition, we plan to investigate methods to update the localization parameters in an adaptive fashion based on the driving environment.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1A6A1A03043144).

References

[1] J. Breßler, P. Reisdorf, M. Obst, and G. Wanielik, "GNSS positioning in non-line-of-sight context—A survey," in *Proceedings of 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1147–1154.

[2] R. W. Wolcott and R. M. Eustice, "Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 292–319, 2017.

[3] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognition*, vol. 74, pp. 90–109, 2018.

[4] H. Sobreira, C. M. Costa, I. Sousa, L. Rocha, J. Lima, P. Farias, et al., "Map-matching algorithms for robot self-localization: A comparison between perfect match, iterative closest point and normal distributions transform," *Journal of Intelligent and Robotic Systems*, vol. 93, no. 3, pp. 533–546, 2019.

[5] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, pp. 2743–2748.

[6] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Proceedings of SPIE 1611, Sensor Fusion IV: Control Paradigms and Data Structures*, 1992, pp. 586–606.

[7] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[8] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[9] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Proceedings of Robotics: Science and Systems (RSS)*, 2014.

[10] G.-P. Gwon, W.-S. Hur, S.-W. Kim, and S.-W. Seo, "Generation of a precise and efficient lane-level road map for intelligent vehicle systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4517–4533, 2017.

[11] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[12] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[13] T. Foote, "tf: The transform library," in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, 2013.

[14] M. A. Al-Khedher, "Hybrid GPS-GSM localization of automobile tracking system," arXiv preprint arXiv:1201.2630, 2012.

[15] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, 1997, pp. 2783–2788.

[16] F. Zhang, H. Stähle, G. Chen, C. C. C. Simon, C. Buckl, and A. Knoll, "A sensor fusion approach for localization with cumulative error elimination," in *Proceedings of 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2012, pp. 1–6.

[17] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," in *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 921–927.

[18] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Proceedings of Robotics: Science and Systems (RSS)*, 2007.

[19] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1078–1086, 2017.

[20] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *Journal of Field Robotics*, vol 24, no. 10, pp. 803–827, 2007.

[21] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, "Evaluation of 3D registration reliability and speed—A comparison of ICP and NDT," in *Proceedings of 2009 IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 3907–3912.

[22] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 829–846, 2018.

[23] Y. Lu, H. Ma, E. Smart, and H. Yu, "Real-time performance-focused localization techniques for autonomous vehicle: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6082–6100, 2022.

[24] I. Skog and P. Handel, "In-car positioning and navigation technologies—A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4–21, 2009.

[25] M. Obst, N. Mattern, R. Schubert, and G. Wanielik, "Car-to-car communication for accurate vehicle localization—the CoVeL approach," in *Proceedings of International Multi-Conference on Systems, Signals and Devices*, 2012.

[26] A. N. Ndjeng, D. Gruyer, S. Glaser, and A. Lambert, "Low cost IMU–odometer–GPS ego localization for unusual maneuvers," *Information Fusion*, vol. 12, no. 4, pp. 264–274, 2011.

[27] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 329–343, 2011.