Map-based Localization for Autonomous Vehicles

Souparna Roy[1], Bhanuprakash S[1], Muhammad Akram A[1], Rajani Katiyar[1]*, Uttara Kumari M[1]

[1]*Department of Electronics and Communication, RV College of Engineering, Bengaluru, Karnataka, India*

Abstract

Map-based localization is a technique for determining precise position and orientation of a mobile agent in a known environment. It uses the sensor measurements such as visual or range sensor data, with the features and landmarks represented on the map. This paper reports map-based localization for a vehicle using the generated datasets. Algorithms like LiDAR Inertial Odometry via Smoothing and Mapping were used to generate the Point Cloud Data while Iterative Closest Point and Normal Distributions Transforms localization algorithms were used to perform map-based localization. The data sets were collected at 20 to 25 km/h vehicle speed, using Velodyne and Ouster LiDARs in a dynamic environment and the same was compared at 23s mark to evaluate the IMU, mapping, and GPS data. Accuracy of GPS in case of Interactive Closest Point was greater. The large step size for Normal Distributions Transform was found to be more computationally efficient as it processes more data per unit time, making it suitable for use in dynamic environments and move-based to map-based localization transition was achieved.

Keywords*: Map-based localization, Point Cloud Data, LIO-SAM, ICP, NDT*

1.0 Introduction

Autonomous vehicles can redefine mobility, improve road safety by incorporating advanced artificial intelligence, sensor technology, and connectivity. They leverage intricate network of sensors, algorithms, and sophisticated control systems to navigate and make decisions autonomously. Autonomous vehicles have the ability to perceive their environment and respond accordingly. The advancements in map-based localization have been driven by breakthroughs in sensor technology, particularly in the fields of computer vision and LiDAR. Visual odometry algorithms employ cameras to track visual features and estimate the robot's motion, while LiDAR-based approaches leverage laser scans to generate 3D maps and perform localization. The fusion of multiple sensors, such as

**Mail address: Bharatish A, Assistant Professor, Department of Electronics and Communication, RV College of Engineering, Bengaluru – 59, Email: rajanikatiyar@rvce.edu.in, Ph: 9036902505*

cameras and LiDAR, has also shown significant potential in improving localization accuracy and robustness.

Map-based localization is needed for high accuracy and reliability in a variety of environments and conditions. This is particularly important in safety-critical applications, such as autonomous vehicles. Another challenge is the need for real-time performance, as autonomous vehicles must be able to process sensor data and update their position estimates quickly and efficiently. However, several challenges remain in the field of map-based localization. The scalability of map-based localization algorithms to large-scale environments is an ongoing area of research. Real-time performance, computational efficiency, and the ability to handle dynamic environments are critical factors for widespread adoption. Also, robustness against changes in lighting conditions, appearance variations, and occlusions pose additional challenges in visual-based localization approaches. Map-based localization in outdoor environments may also be influenced by seasonal changes, environmental deformations, or limited GPS availability.

Map based localization for autonomous vehicles utilizing the LIO-SAM (LiDAR Inertial Odometry via Smoothing and Mapping) framework in combination with Velodyne and Ouster LiDAR sensors is reported [1-3]. To build extremely precise and economical maps for autonomous vehicles, the LIO-SAM framework includes Simultaneous Localization and Mapping (SLAM) methods.

The use of modern LiDAR sensors, such as Velodyne and Ouster, enables accurate and detailed perception of the surroundings. The strengths, limits, and applicability of NDT and ICP algorithms are examined [4]. The research also explores the implementation of the NDT (Normal Distribution transform) and ICP (Iterative closest point) algorithms. The NDT technique makes use of a statistical model to characterize the surrounding area and calculate the pose of the vehicle, whereas the ICP approach iteratively matches the observed point cloud data via the map data to improve the localization estimation.

The LIO-SAM method, developed by Shan et al. [1] reliably locates a vehicle in real-time by combining readings from inertial sensors and LiDAR odometry. The system performs exact pose estimation and retains a small number of keyframes while using the high-resolution 3D point cloud data from LiDAR sensors and a sliding window approach for simultaneous localization and mapping, assuring accuracy and efficiency. By offering an accurate approach for applications like mapping, object identification, and autonomous navigation, this method advances the area

of map-based localization for autonomous vehicles. Hossain and Lin proposed a method called Uncertainty-Aware Tightly-Coupled GPS Fused LIO-SLAM [2] for autonomous delivery vehicles operating in areas where GPS access is limited or denied. Through the combination of GPS and LiDAR inertial odometry in a tightly-coupled manner and the use of an Extended Kalman filter for fusion, this method effectively overcomes long-range drift issues and produces accurate maps in outdoor as well as semi-indoor/indoor environments. The experimental results demonstrated its effectiveness by achieving lower RMSE than those of the GPS based mapping approaches.

Wang et al. [3] utilized 3D-LiDAR information to deliver a map-based localization approach for autonomous vehicles. The approach precisely determines the vehicle's location and orientation relative to the map by merging an already constructed map with real-time LiDAR data.

The ICP technique is used to build a high-resolution map, while SVD is used to estimate pose. The usefulness of the approach for attaining exact localization in different contexts is demonstrated by empirical results using real-time data from autonomous cars.

Shin et al. [4] discussed a high-definition map localization technique for self-driving cars that makes use of advanced driver assistance system (ADAS) environment sensors. The technique contains parts for representing environment features, digital map analysis, map-based position correction, predefined validation gates, and extended EKF-based localization filtering and fusion. The system detects lane information with monocular vision and the quad rail with radar using numerous feature extraction stages.

Chen et al. [5] presented NDT-LOAM, a real-time LiDAR odometry and mapping method that analyses LiDAR point cloud data for simultaneous localization and mapping. To improve the speed of computation, the framework separates LiDAR SLAM into front-end odometry and back-end optimization. To eliminate accumulated mistakes, the front end employs the Normal Distribution Transform (NDT) for point cloud registration. Weighted NDT in combination with Local Feature Adjustment, building novel cost functions having weights depending on range values and surface features are used to analyse point clouds. NDT-LOAM beat state-of-the-art algorithms like ALOAM/LOAM on the KITTI dataset, displaying superior translation drift (0.899% average) and real-time performance (10Hz). The results show that NDT-LOAM is a highly precise, low-drift approach for LiDAR-based mapping and localization.

Liu et al. [6] suggested an NDT-based real-time high-precision positioning and mapping method for large scenes. The method solves LiDAR registration difficulties by enhancing stability, accuracy, and vehicle positioning. It demonstrates excellent durability, enhanced monitoring performance, and accurate positioning even in situations of signal fading by employing NDT properties and incorporating them into the SLAM framework.

The objective of this research was to achieve map-based localization for a vehicle using the generated datasets. Algorithms like LiDAR Inertial Odometry via Smoothing and Mapping were used to generate the Point Cloud Data while Iterative Closest Point and Normal Distributions Transforms localization algorithms were used to perform map-based localization.

2.0 Methodology

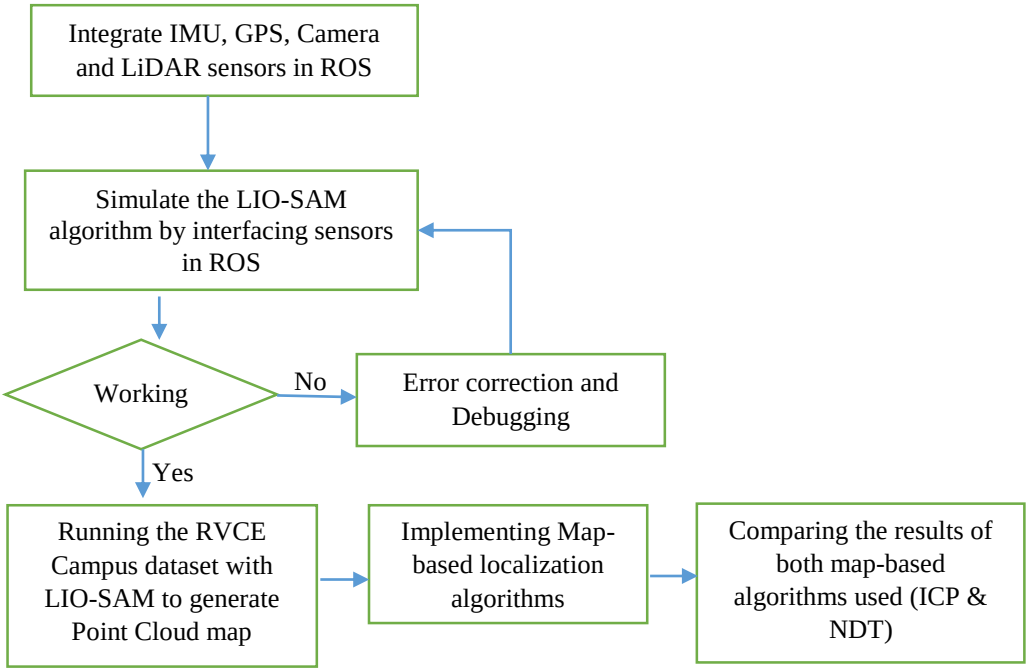

Fig. 1. Methodology for Map-based Localization for Autonomous Vehicles

Fig. 1 shows the methodology adopted for the design and implementation of Map-based localisation for autonomous vehicles. The methodology involves installation of Robot Operating System (ROS), setting up of ROS environment, installation of the drivers for IMU, GPS, and LiDAR and

their integration with ROS to provide a platform for running LIO-SAM. Once the LIO-SAM has successfully generated the required pcd maps, the implementation of map-based localization can be started. Two map-based localization algorithms, ICP and NDT Localization, were used and the results were compared. The algorithms were tested for public and local datasets. The robot's status and trajectory was calculated using the sensor data of 3D LiDAR, IMU, and GPS. This state estimation problem was formulated as a map posterior problem.

A factor graph was used to model the problem. With no loss of generality, the system can incorporate measurements from other sensors like heading from a compass or altitude from an altimeter. For the purpose of building factor graphs, four different kinds of factors and one type of variable were introduced. The nodes of the graph are responsible for this variable, which captures the state of the robot at a particular moment.

The four basic factors used are:

i. IMU Pre-integration Factors: It focuses on finding the relative motion between two consecutive timesteps and hence, the pre-integrated time measurements Δvij (velocity), Δpij (position), ΔRij (rotation) between the time i and j [7]:

$$\Delta vij = RiT\ (vj - vi - g\ \Delta tij) \quad (1)$$

$$\Delta pij = RiT\ (pj - pi - vi\ \Delta tij - 1/2g\ \Delta tij2 \quad (2)$$

$$\Delta Rij = RiT\ Rj \quad (3)$$

ii. LiDAR Odometry Factors: Key-frames were introduced as the computation for individual LiDAR frames and adding them to the factor graphs is not efficient. In this approach, a LiDAR frame Fi+1 is selected as a keyframe when the change in the pose of the robot exceeds a user-defined threshold when compared to the previous state x. The new key frame, Fi+1, is associated with a new node of the robot, xi+1 in the factor graph. The LiDAR frames between these two keyframes are discarded. The addition of keyframes in this manner helps to achieve a balance between memory consumption and map density while also maintaining a relatively sparse factor graph. This is suitable for real-time nonlinear optimization. The position and rotation thresholds for adding a new keyframe are chosen as 1m and 10°.

iii. GPS Factors: It is used to eliminate drift when performing long-term navigation tasks. Sensors that provide absolute measurements to eliminate drift can be introduced. These sensors consist of a GPS, compass, and altimeter. The GPS readings are first translated into the

local Cartesian coordinate system and then a new GPS factor with this node is associated when a new node is added to the factor graph. If the GPS signal and the LiDAR frame are not hardware synchronized, linear interpolation between GPS measurements based on the timestamp of the LiDAR frame is performed. Since the drift of LiDAR inertial odometry increases relatively slowly, it is not required to continuously apply GPS components when GPS reception is available. In actual use, only a GPS component is used when the projected position covariance exceeds the GPS position covariance that was received.

iv. Loop Closure Factors: It implements a simple Euclidean distance-based loop closure detection method. For the first search of the factor graph, the prior states that are close to the new state xi+1 in Euclidean space whenever a new state xi+1 is added. The Frames are matched to the sub-frames by using scan-matching. The search distances for loop closures are set to be 15m from a new state xi+1 and the index m to be 12. However, in practice when GPS is the only absolute sensor available, it is found that adding loop closure factors is very helpful for reversing the drift in a robot's height. This is due to the extremely unreliable height measurement provided by GPS, which led to altitude errors nearing $100m$ in the absence of loop closures.

ICP [8] algorithm: It is used for global position estimation. In a process known as scan registration, a source point cloud is aligned to a set target point cloud by determining the best possible translation and rotation to reduce the distance between the two. There are other ICP variants, such as point-to-plane, to have better convergence than that of the point-to-point.

Point-to-plane variation of ICP: ICP's point-to-plane variation is frequently more effective than the traditional point-to-point kind at determining the ideal transformation. By estimating the surface normals, for each of the points of the target cloud, the tangent planes are created. The surface normals for each point in O are denoted by nj.

Sources of error: When implementing ICP, a few causes of error must be taken into account. ICP converges incorrectly to a local minimum rather than a global minimum. Since ICP does not account for this while minimizing the cost function, this frequently turns out to be the dominant error. Before using ICP, an initial alignment can be done to get the source cloud closer to the global minimum and prevent this problem. Either a qualified estimate or a random selection of different initial positions can be used to accomplish this. Before each iteration of the ICP algorithm, a random Gaussian noise translation is introduced to the source cloud to enable the algorithm to relocate from a local minimum.

The second source of error is under-constrained circumstances, such as those in which there is insufficient data to determine the location and orientation of the object in question. Since the ICP is unable to identify the exact location of the object, it can be seen that sliding takes place in one of the coordinate axes in the various findings.

The two other concepts necessary for ICP are:

Iterative Closest Point: The results of the pre-processing step are the input data used in the ICP algorithm. In other words, the target point cloud, which in this system is similar to an elevation map of the area, and the pre-processed source point cloud, which was created from the LiDAR sensor mounted on the vehicle. A subset of points are retrieved. In this stage, the entire cloud population, as well as this subset, is utilized.

Uncertainty Estimation: Finding the uncertainty of the result produced by ICP is the second and final phase of the global position estimate procedure, which is accomplished by estimating the covariance. The location estimations provided by ICP are weighted by the covariance in the EKF (Extended Kalman Filter). The UT-based covariance estimate, covariance with correspondences, and covariance with Hessian are the three covariance approaches that are put to the test.

The covariance is estimated in three dimensions because the point clouds are three-dimensional, but only the x, y, and θ (rotation around the z-axis) elements of the covariance matrix are chosen and put into a three-dimensional matrix. This is due to the fact that the EKF can only estimate 2D positions because the local position estimations generated by the present FOI system's odometry data are in 2D. After the ICP technique, after the source cloud has been aligned to the global coordinate system the covariance is computed.

3.0 NDT Algorithm

It is used for probabilistic scan matching and registration of point clouds. It is commonly employed in map localization, simultaneous localization and mapping (SLAM), and object recognition tasks. NDT is known for its robustness to noise and outliers, making it suitable for environments with sensor uncertainties.

The NDT algorithm represents point clouds as a probabilistic distribution, specifically a Gaussian distribution. The steps involved in the NDT algorithm are as follows:

Voxelization: Voxelization is the process that converts a data structure, storing geometric information in a contiguous domain, into a rasterized

image. The point clouds are divided into equally sized voxels to create a voxel grid representation. Each voxel represents a local region of the environment.

Feature Extraction: Feature Extraction is the process by which an algorithm extracts a set of informative and non-redundant data to help the learning for the algorithm and generalize the further steps.

Initially, an estimate of the relative pose between the target point cloud and the reference map is obtained using an initial guess, such as odometry data. For each voxel in the target point cloud, the closest voxel in the reference map is found. Assigning weights to the correspondence is performed based on the similarity measure.

High weights are given to voxels with similar distributions, while low weights are assigned to dissimilar voxels or outliers.

Using the correspondence and their weights, an optimal transformation matrix that aligns the two-point clouds is estimated. The estimated transformation is applied to the target point cloud, aligning it with the reference map. The algorithm checks if the estimated transformation matrix has reached the desired accuracy or if the change between iterations falls below a specified threshold. If not, the algorithm returns to the final transformation matrix and provides the optimal alignment between the target point cloud and the reference map, representing the relative pose estimation.

4.0 Results and Discussion

Fig 2 shows the setup of autonomous vehicle on which the Velodyne and Ouster LiDARs.

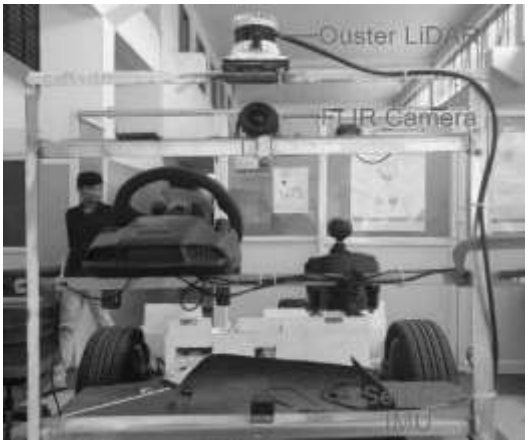

Fig. 2. Autonomous vehicle set up with Ouster LiDAR, FLIR Camera and XSens MTi IMU

The data pertaining to the sensors were verified. The Linear Acceleration data from both LiDARs were compared and found to be agreeing as shown in Fig. 3.

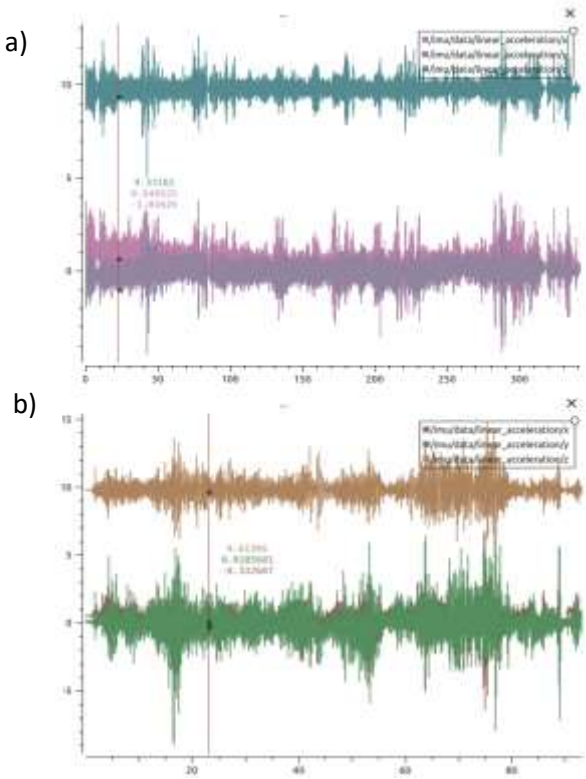

Fig. 3 a) Linear Acceleration data from Velodyne Sensor
b) Linear Acceleration data from Ouster Sensor

LIO-SAM algorithm was used to map the environment and generate a pcd map for the same. It enabled localizing the autonomous vehicle and made the self-guided movement possible. The red color shows the area that is being mapped with higher intensity while the green color towards the left indicates mapping with a lower intensity due to the range constraints of the LiDARs.

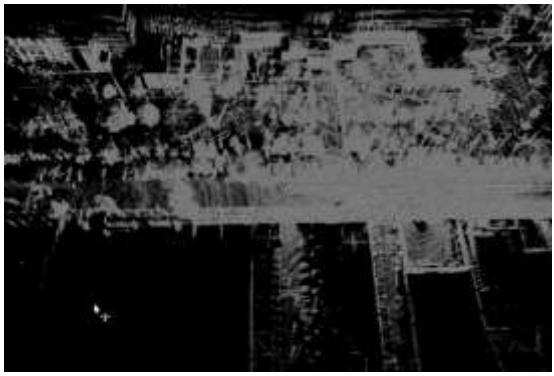

Fig. 4. LIO-SAM execution in the environment

Fig. 5 shows the pcd map generated using LIO-SAM for the environment as well as the corresponding google maps image for the place using Velodyne LiDAR. Fig. 6 shows the same using the Ouster LiDAR. The google maps image clearly shows the environment mapped and the pcd files helps visualizing the data points for the same in 3D.

Fig. 5. Google maps, pcd map with Velodyne LiDAR

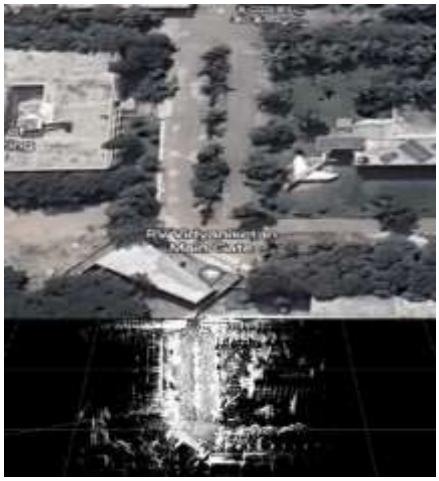

Fig. 6. Google maps, pcd map with Ouster LiDAR

The maps shown in Fig. 5 and Fig. 6 were used to perform map-based localization\cite{p9}. For the ICP localization, the pcd map was given as input to the algorithm. The map gets uploaded to the background and can be seen from the grey data points in Fig. 7. The bright green circle in the middle of the RGB area indicates the position of the vehicle fitted with the LiDAR sensor. The concentric circles around the central position indicate

the field of view of LiDAR and its effective range. The changing color from green to blue toward the outer edges indicates that the points that are furthest away from the LiDAR are being recorded with a lesser intensity.

Fig. 7. ICP Localization with RVCE Campus Dataset

The output estimates the vehicle's pose, which provides information about its position and orientation relative to the pre-built map. This pose estimate is continuously updated as the vehicle moves through the environment, allowing it to determine its location accurately.

Fig. 8 shows NDT localization where the pcd map and its corresponding csv file [10] are given as input to the algorithm. The map can be seen as a set of white data points in the background. The RGB axis towards the bottom indicates the starting position for the vehicle and also the starting point for localization. The RGB axis towards the right of the figure indicates the current position of the vehicle that is coming back to close the loop at the starting point. The RGB points all over the map indicate the localization that is happening via the algorithm.

Fig. 8. NDT Localization with Dataset

The NDT algorithm typically provides a covariance matrix representing the uncertainty or confidence associated with the estimated location. This covariance matrix describes the level of uncertainty in estimates of position and orientation. The results using the environment datasets estimate the location of the vehicle, including its position and direction on the reference map.

5.0 Conclusion

A map-based localization system was developed by using lidar odometry and multi-sensor fusion. It was deployed to perform localization in a select environment. Frameworks like LIO-SAM were used to provide initial lidar inertial odometry, to map and generate pcd maps for the environment. The maps were utilized in algorithms like ICP and NDT to perform map-based localization. The autonomous vehicle was successfully localized based on the data gathered by the sensors in real time and the path data available from the 3D map. The method was tested using multiple public datasets and the datasets collected in the environment.

The Velodyne and Ouster LiDARs had a run time of 345.71s and 93.18s respectively while collecting datasets. Both the datasets were compared at the 23s mark to evaluate the IMU, mapping, and GPS data and ensure uniformity of input for both the algorithms. The GPS accuracy in case of ICP was greater. For ICP, the units of visualization used were FlatSquares with a size of 0.15m and for NDT, the same were FlatSquares with a size of 3m. The large step size for NDT showed that it is more computationally efficient as it processes more data per unit time. The results showed that NDT is better suited to localization in a dynamic environment but ICP is better when multiple sensors have to be changed frequently. A combination of NDT and ICP helped in creating a functional system and transition from move-based to map-based.

References:

1. Shan, Tixiao, Lio-sam, Tightly-coupled lidar inertial odometry via smoothing and mapping, *IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE*, 2020.
2. Hossain, Sabir, and Xianke Lin, Uncertainty-Aware Tightly-Coupled GPS Fused LIO-SLAM, *arXiv preprint arXiv*:2209.10047, 2022.
3. Wang, Liang, Yihuan Zhang, and Jun Wang, Map-based localization method for autonomous vehicles using 3D-LIDAR, *IFAC-PapersOnLine* 50.1: 276-281, 2017.

4. Shin, Donghoon, Kang-moon Park, and Manbok Park, High definition map-based localization using ADAS environment sensors for application to automated driving vehicles, *Applied Sciences* 10.14: 4924, 2020
5. Chen, Shoubin, et al., NDT-LOAM: A Real-time Lidar odometry and mapping with weighted NDT and LFA, *IEEE Sensors Journal,* 22.4: 3660-3671,2021.
6. Liu, Sijia, et al., Research on NDT-based positioning for autonomous driving, *E3S Web of Conferences EDP Sciences,* 257, 2021.
7. Forster, Christian, Luca Carlone, Frank Dellaert, and Davide Scaramuzza, On-manifold preintegration for real-time visual-inertial odometry, *IEEE Transactions on Robotics* 33(1), 1-21, 2016.
8. Nylén, Rebecka, and Katherine Rajala, Development of an ICP-based Global Localization System, 2021.