THE UNIVERSITY OF WESTERN AUSTRALIA
Achieve International Excellence

GENG5512 Final Year Research Project Thesis

LiDAR-based Simultaneous Localization and Mapping System for nUWAy

Yuchen Du (22202224)

(word count: 6367)

School of Electrical, Electronic and Computer Engineering
The University of Western Australia

Supervisor: Professor Dr Thomas Bräunl

Robotics and Automation Lab
The University of Western Australia

Abstract

Simultaneous localization and mapping technologies play an important role in the autonomous driving task. Laser-based systems are preferred choices among a number of solutions to the SLAM problem due to its capability of giving a robust result in outdoor environments. This thesis firstly introduces the project background and project scope, and then summarizes the issues existing in the practical SLAM scenario, that is, the scenario related to the nUWAy project and the terrain of the UWA campus. After that, the author further investigates the recent work related to Lidar-based SLAM, focusing on the presented approaches, achieved outcomes and how these can be used for problems in the nUWAy project. According to the design requirements, a SLAM framework is proposed underpinned by Google cartographer. The mapping result for the UWA main campus and further discussion demonstrates this expanded approach partially addressed the identified problems in this specific scenario.

Nomenclature

UWA	The University of Western Australia
REV	Renewable Energy Vehicle
nUWAy	UWA autonomous bus
SLAM	Simultaneous Localization and Mapping
GPS	Global Positioning System
IMU	Inertial Measurement Unit
ROS	Robot Operating System
PCL	Point Cloud Library
KITTI	A popular benchmark autonomous driving dataset
CARLA	Open-source simulator for autonomous driving research
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
LOAM	Lidar Odometry and Mapping
SBPL	Search-based path planner
RTK	Real-time kinematic

Table of Contents

Abstract 2
Nomenclature 3
1. Introduction 5
1.1 Overview 5
1.2 Project background 6
1.3 Project scope 7
2. Problem identification 8
3. Literature survey 10
4. Design process 16
4.1 Requirements 16
4.2 Methodology 17
4.3 Evaluation 17
4.4 Design tools 18
5. Final design 22
5.1 Framework 22
5.2 Results 24
5.3 Discussion 26
6. Future work 28
7. Conclusion 29
8. Reference 30

1. Introduction

1.1 Overview

With the rapid development of automobile electronization and advanced assisted driving technology, autonomous driving, as an advanced stage of assisted driving technology, has become an important way to solve transportation problems in the future, and has become a new research focus in the world. Self-driving vehicles have the potential to reshape mobility by enhancing the safety, accessibility, efficiency, and convenience of automotive transportation.

One task of a self-driving vehicle is to navigate itself from an initial place to the destination autonomously. This task normally requires the map of the environment where the autonomous vehicle operates, and the localization information within this map. Figure 1[1] is a typical architecture of the autonomy system of self-driving cars, which demonstrates the importance of the localization information and the map of the operation area. This system consists of two components, a perception system and a decision-making system. The path planner in the decision-making system is able to generate a path to a given destination without hitting any obstacle based on the drivable areas recorded in the map. To get the car to move along the path, the localizer should estimate where the vehicle is timely. Based on the localization information from the localizer and the path created before, the motion planner can accordingly derive the correct control command which allows the self-driving car to follow the path.

In order to obtain the map required by the autonomous driving system, the mapping process shall record environmental features within a coordinate frame. If the accurate poses (position and orientation) of the vehicle at any time are known, there are various and effective methods to build the map[3][4][5]. Although GPS can give the pose information, its signal cannot be guaranteed “in occluded areas, such as under trees, in urban canyons (roads surrounded by large buildings) or in tunnels."[1] In these circumstances, the vehicle usually cannot get its position and orientation directly when it is operating in an unknown environment. This requires autonomous vehicles to estimate the poses using onboard sensors while driving, and incrementally create the map at the same time. In other words, drive-less cars need simultaneous localization and mapping (SLAM) technology to address these issues.

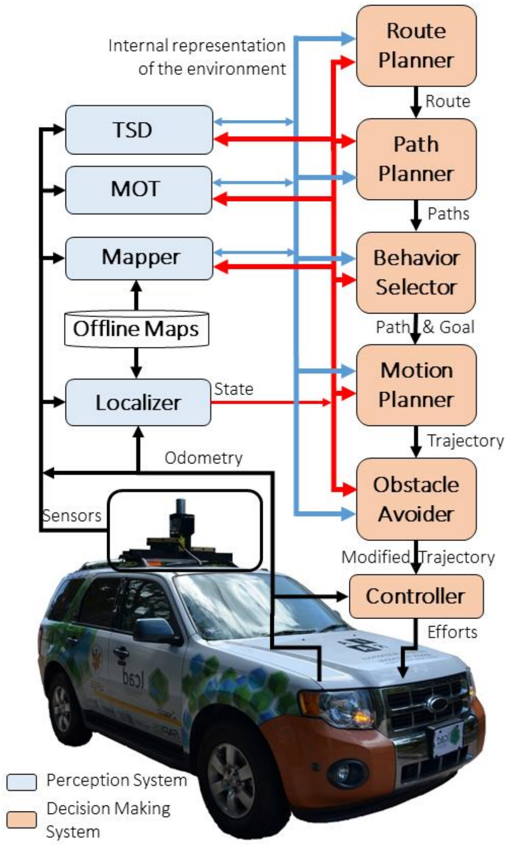

Figure 1. Overview of the typical hierarchical architecture of self-driving cars[1]

1.2 Project background

The nUWAy is a newly launched project by the Renewable Energy Vehicle team (REV) of the University of Western Australia (UWA), aiming to integrate autonomous driving technologies developed by REV to an electric shuttle bus platform without any of the high-level driving and navigation systems from its manufactory EasyMile[16]. The goal of nUWAy autonomous bus focuses on smooth driving around roads between locations in the Crawley campus of UWA, which requires a map for global path planning. Also, this EZ10 electric shuttle bus platform is equipped with various LiDARs which is able to cover almost all the surrounding environment as shown in Figure 2[6]. Based on this situation, the Lidar-based SLAM scheme is given a higher priority to assist the task of nUWAy bus because of the strong performance of these laser range finders.

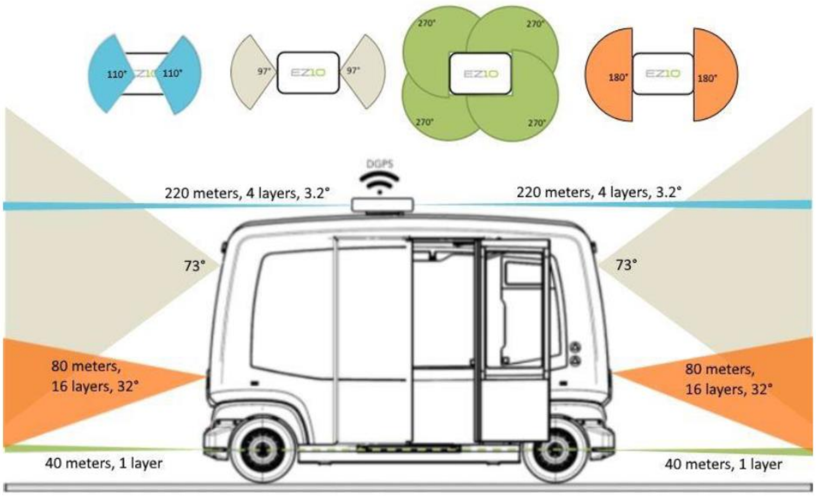

Figure 2. Summary of the EZ10 V2.3 localization and detection technologies[6]

1.3 Project scope

As mentioned before, SLAM plays an important role in autonomous driving. This research is intended to supply a SLAM system suitable for our nUWAy project in Crawly campus. To be specific, it provides an offline grid map of the campus essential for path planning, and the localization information during navigation.

2. Problem identification

Laser-based systems are preferred choices among a number of solutions to the SLAM problem due to its capability of giving a robust result in outdoor environments[7]. However, when applying the Lidar-based solution to nUWAy project some difficulties arise.

As mentioned previously, several Lidars are mounted on the autonomous shuttle bus, including four SICK 2D Lidars, two Velodyne 3D Lidars and two Ibeo Lux Lidars. From Figure 2, it can be observed that SICK Lidars only scan the environment at a lower height while Ibeo Lux Lidars do it at a higher height. If only lidars at one altitude are used for SLAM, the obstruction at other height would not be recorded in the map. The path planner may thus treat a place with obstacles as a possible route. Therefore, under the condition that a total of eight Lidars are loaded on nUWAy, the way how these 2D and 3D Lidars are combined to gain a better grid map result is the first problem.

Also, in the process of SLAM, the map is gradually created as the autonomous vehicle moves in the environment, so the accurate pose estimation holds the key to the mapping process. However, if the pose estimation only relies on the frontend odometry, the drift issue comes about, which refers to the accumulation of the estimation errors. Concretely, the frontend odometry estimates the current pose based on the pose of one previous step, so the error in earlier estimated pose would accumulate as the estimation process goes. In other words, the longer the driving time is, the more distorted the map will be. Therefore, how the drift phenomenon is handled is another issue.

In addition, some certain circumstances make pose estimation more difficult. For example, James Oval in UWA campus is an open space which lacks geometric features. If the vehicle is on the road across the oval, it hardly collects the information of the surroundings using lidars since most of lasers would not be reflected from the open space. This lack of the environmental information makes it difficult to estimate vehicle's trajectory from observing the landmarks. Also, the slopes in the environment would make negative impacts on the estimation result if the vehicle only depends on the lidar data. More specifically, the vehicle with solely lidar data available cannot distinguish whether it's moving on a flat ground or a slope. Accordingly, these circumstances need to be taken into account as well.

Furthermore, many subsystems in autonomy system need real-time localization information, such as the motion planner that calculates commands to control autonomous cars to operate as expected. This requires the localization system to provide required pose information in real-time as well. Nevertheless, general SLAM technologies consume high computing resources so the mapping process is normally performed offline. Therefore, configuring the SLAM technology to achieve online localization is a key problem in autonomous driving.

3. Literature survey

"Simultaneous Localization And Mapping (SLAM) has been a hugely popular topic among the mobile robotics community for more than 25 years now"[8]. Lidar-based SLAM is one of the most reliable SLAM solutions in this field as it is not sensitive to the changes in light conditions and can give robust outcomes compared to SLAM approaches based on other sensors. This section reviews recent work related to Lidar-based SLAM, focusing on the presented approaches, achieved outcomes and how these can be used for problems stated in the last section.

Grisetti G et al. presented Gmapping based on Rao-Blackwellized particle filters in 2007[9]. In traditional Rao-Blackwellized particle filters, the large number of particles required to build an accurate map consumes a lot of memory resources, and particle impoverishment effect arises as the resampling step can potentially eliminate the correct particle. In order to solve the serious memory consumption problem and keep the number of particles in a relatively small number, Gampping samples a proposal distribution, and then optimizes the pose based on the optimized scan-matching. For relieving the particle depletion problem, the approach sets a threshold to determine whether the resampling is performed or not. When the number of particles exceeds the threshold, it stops resampling, otherwise performs resampling again. This approach reduces the number of unnecessary resampling actions in the particle filter and thus slows down particle depletion. Gmapping is the most used SLAM package in robots, and it can be used for indoor and outdoor environment. However, as the scene grows, the number of particles increases and the requirement for memory and computational resources go up as well, so it's not suitable for building a large map. Also, due to the lack of loop closure detection, the map may be misaligned when a visited place is reached. Therefore, Gmapping is hardly applied to the nUWAy project.

Kohlbrecher S et al. introduce Hector SLAM for Urban Search and Rescue scenarios in 2011[11]. This system provides a loosely coupled IMU-Lidar SLAM solution, and it consists of two relatively independent subsystems as shown in Figure 3[11]. 2D SLAM subsystem provides position and heading information within the ground plane. It only relies on scan-matching algorithm, which estimates translation and rotation of the robot between two adjacent Lidar scans. The scan-matching problem is solved by Gaussian-Newton method and this may result in a local minimum. In order to achieve global optimality, multi-resolution maps are considered. 3D navigation subsystem supplies 3D motion estimation based on IMU. Extended Kalman Filter is used to improve IMU drifting by fusing other sensor data such as GPS. This 3D estimate result can also contribute to the performance of 2D SLAM subsystem by projecting pose estimate of the EKF on the xy-plane as start estimate for the optimization process of the scan matcher. The main advantage of this approach is that it requires low computational resources, thus there is no loop closure detection in the process. Therefore, it may not work well for a large scenario.

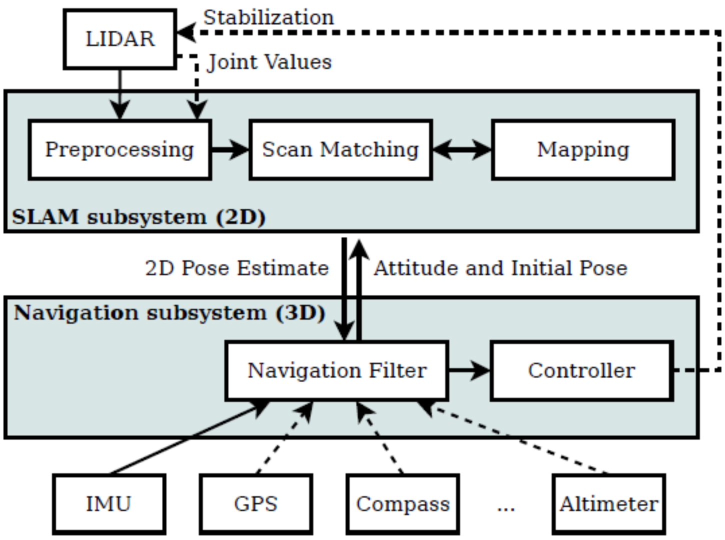

Figure 3. Overview of the mapping and navigation system (dashed lines depict optional information) [11]

Google Cartographer SLAM system is proposed by Hess W et al. in 2016[12]. The main contribution of the approach is to achieve real-time localization and mapping with loop closure detection at 5cm resolution. The system consists of local SLAM and global SLAM as depicted in Figure 4[14]. In order to reduce accumulative error in scan-matching step of local SLAM, a graph-based pose optimization is adopted in global SLAM.

The SLAM process is listed as follow:

1. IMU gives a good start estimate to the Ceres-based scan matcher in local SLAM, so that the current pose can be estimated and the corresponding scan is inserted into the local submap;
2. After a submap finished, it along with previous finished submaps would takes part in loop closure;
3. If the current pose given by local SLAM is close enough to some submaps, i.e. visited places, a scan matcher in global SLAM tries to find the corresponding observation in the submap. In this step, a branch-and-bound approach is used to speed up scan-to-submap matching in loop closure process;
4. If a good match is found in a search window around the currently estimated pose, it is added as a loop closing constraint to the optimization problem;
5. Based on constraints from previous steps, we run a pose optimization every few seconds so that an accurate localization and mapping can be achieved.

This technology enables us to mapping the very large environment up to tens-of-thousands of square meters, which may be applied to our project. This system is able to fuse data from multiple sensors, which fits the case of the shuttle bus with multiple Lidars.

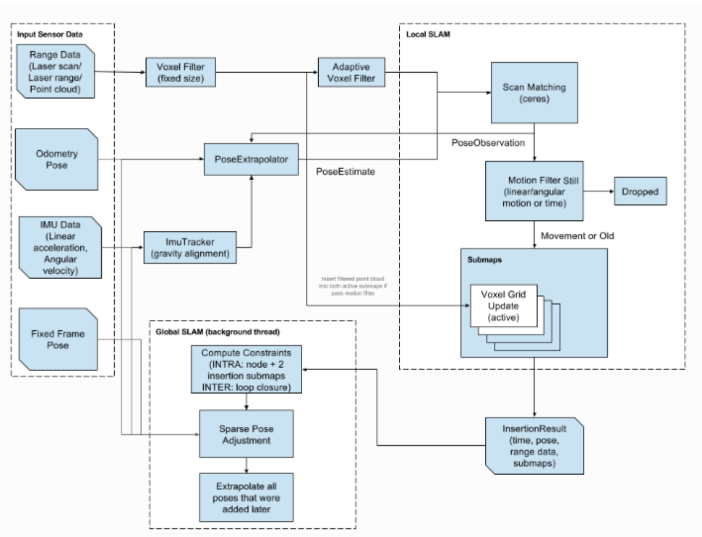

Figure 4. Overview of Google Cartographer[14]

In 2018, Ji Zhang et al. presented a real-time SLAM algorithm called LOAM, that is, Lidar Odometry and Mapping. LOAM method performs the frontend odometry which gives a low-drift result with low-computational complexity. This outcome is achieved by dividing the complicated SLAM problem[17], which seeks to optimize a large number of variables simultaneously, into motion estimation problem and mapping problem. As shown in Figure 5[13], one algorithm running at a high frequency is intended to estimate the motion of the Lidar. This fast-running algorithm is able to reduce the effect of mis-registration of point cloud caused by motion of Lidars, so that it allows accurate motion estimation. Another Lidar mapping algorithm runs in parallel with the motion estimation algorithm and performs fine processing to create maps at a lower frequency. Combination of the two algorithms results in low-drift motion estimation and mapping in real-time. Also, the scan-matching part of these two algorithms are based on feature points instead of normal scan-to-scan approach, which reduces computational complexity.

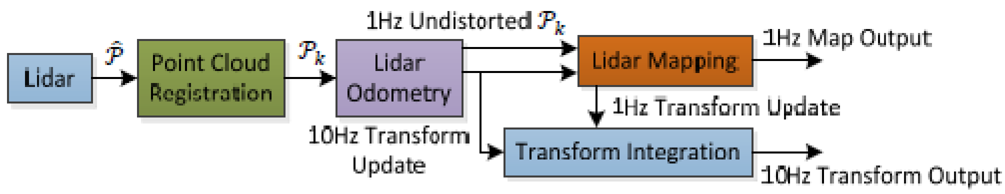

Figure 5. Block diagram of the lidar odometry and mapping software system[13]

Tim Stahl et al. introduced an improved Adaptive Monte Carlo Localization (AMCL) method for a race vehicle at high-speed in 2019[18]. Several challenges shall be addressed in this racing scenario. Firstly, the low computation time is essential for the race vehicle to locate itself at high velocities. Without a decent update frequency, the high-speed vehicle would lose its position and orientation over a large distance even though it drives for a short period of time. Another one is that inaccurate localizations at high velocities should be detected and handled timely to avoid severe results. In order to improve computation time, one way is to reduce the amount of the Lidar beams processed in measurement update of AMCL. The default AMCL algorithm samples

Lidar measurements at an equal angular displacement from all the Lidar beams. In the racing track, "most of the extracted beams hit the wall right next to the vehicle in a similar spot, which does not provide a substantial information gain," as shown in Figure 6. This improved approach determines the angular displacement based on equal distances on the borderline of a surrounding rectangle, which can obtain more information with less amount of Lidar beams.

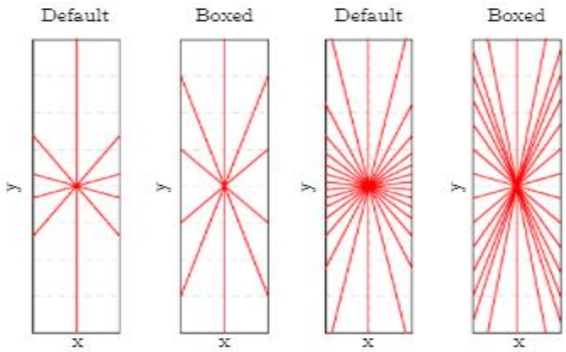

Figure 6. Default measurements and improved measurements[18]

Apart from this, the original AMCL generates a large spread in particles to represent possible poses. However, high velocities limit the change in orientations of consecutive poses, since the upper bound of the lateral tire force is easily exceeded at high speeds. The proposed approach controls the spread of the particles in a velocity-dependent way so that generated particles maintain the same heading tendency, as illustrated in Figure 7. Therefore, the elimination of unlikely particles results in a lower computational effort. For detecting the inaccurate localization estimate, the paper introduces the introspective performance evaluation which generates a localization status, a map dependent status and the variance of the pose estimation to indicate the validity of the result.

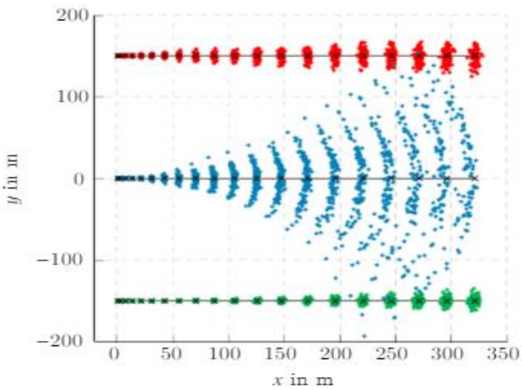

Figure 7. Hedaing tendency for original and modified methods[18]

Mariano Jaimez et al. in 2016 proposed a range flow-based approach to implement a planar odometry from radial laser scanner[19]. Based on the Taylor expansion and the rigidity hypothesis, the range flow constraint equation shown below is transformed into a constraint for the lidar velocity:

$$\left(\cos\theta + \frac{R_\alpha k_\alpha \sin\theta}{r}\right) v_{x,s} + \left(\sin\theta - \frac{R_\alpha k_\alpha \cos\theta}{r}\right) v_{y,s} \\ + \left(x\sin\theta - y\cos\theta - R_\alpha k_\alpha\right)\omega_s + R_t = 0$$

From this equation, it can be observed that only 3 independent restrictions are sufficient to estimate the 2D sensor velocity $\xi_s = (v_{x,s}, v_{y,s}, \omega_s)$. However, due to the noise from the measurements and the errors caused by linearization approximation, this velocity estimation problem in this paper is transformed into an optimization problem which tries to minimizing all the geometric residuals. In order to further overcome these limitations, the pre-weighting strategy and the coarse-to-fine scheme are adopted to compute the optimal motion estimation. “Compared to point-to-line iterative closest point (PL-ICP)[27] and the polar scan matching approach (PSM), the RF2O approach demonstrates the superior performance”[19], as shown in Figure 8. In our nUWAy project, this method can be considered as the front end lidar odometry in the lidar-based localization system.

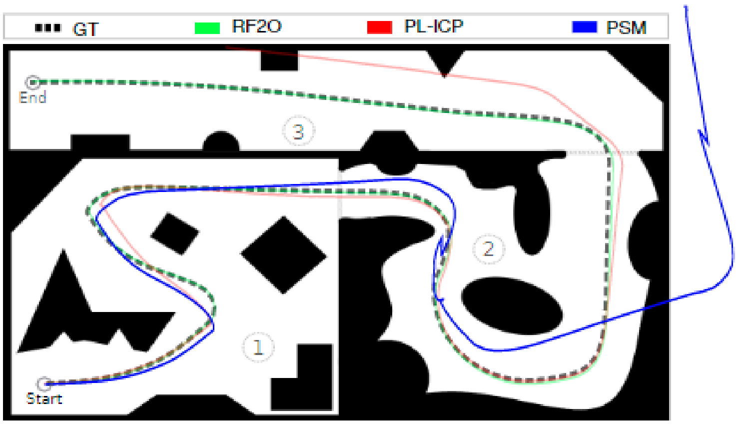

Figure 8. Comparison among three different laser odometry[19]

4. Design process

This section introduces four requirements of the mapping and localization system in nUWAy shuttle bus, the design method used to arrive at the final design and the design tools that contribute to the system.

4.1 Requirements

As mentioned in section 2, four requirements are summarized for this design.

1. The obstacles that may affect the operation of the nUWAy bus in the environment may be at different height, and the lidars at one level are likely to miss these obstacles at different height. In order to record all the information in the map, all eight lidars need to be used for perception. The meriging of the point cloud data from all lidars is the first requirement.

2. Only using frontend odometry (dead reckoning) in the mapping process results in the drift issue which would cause the distortion in the created map. The final design should be able to handle the accumulated error.

3. Certain circumstances such as open spaces and slopes negatively impact the pose estimation from the frontend odometry if the state estimation only depends on lidars. Combining other sensors suck as GPS or IMU in the estimation process is the third requirement.

4. Autonomy system requires real-time localization information for driving the vehicle safely. However, general SLAM technologies need high computing resources. When it is running with other subsystems, the localizer may give a state estimation with a time delay, giving rise to severe safety issues. Thereby, the fourth requirement is to reduce the computational load for the localization system.

4.2 Methodology

The nUWAy project is launched at the beginning of 2020, so the first stage of this project focused on the hardware configuration for the nUWAy bus. Since shuttle bus cannot operate in the campus, the main goal at this stage is to research present SLAM algorithms and evaluate their performance. Several SLAM algorithms are open source and integrated as ROS packages, including Gmapping, Karto SLAM, Hector SLAM and Google Cartographer. Also, an alternative method for localization is 'robot_localization' ROS package, which provides an EKF node for fusing data from IMU, GPS and odometry[15]. The performance of these algorithms can be tested on the KITTI dataset and CARLA simulator.

Once the low-level work is done, it is possible to drive the bus for collecting the sensor data from the campus. According to the requirements and the landscape of the campus, the researched SLAM and localization methods may be expanded and refined to fit in the real scenario.

4.3 Evaluation

Two evaluation metrics are listed below:

A. Relative pose error (APE)

"The relative pose error measures the local accuracy of the trajectory over a fixed time interval Δ."[20] In other words, the drift of the trajectory can be described by this metric. Therefore, APE contributes to the evaluation of frontend odometry. The relative pose error at time step i is defined as

$$E_i := (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta})$$

Where P is the estimated trajectory and Q is the ground truth trajectory.

The root mean squared error (RMSE) over all time steps for the translational components can computed by

$$\mathrm{RMSE}(E_{1:n}, \Delta) := (\frac{1}{m}\sum_{i=1}^{m}\|trans(E_i)\|^2)^{\frac{1}{2}}$$

B. Absolute trajectory error (ATE)

The absolute trajectory error can be used to measure the global consistency of the estimated trajectory. The definition of ATE is given by

$$F_i := Q_i^{-1} S P_i$$

Where S is a rigid-body transformation corresponding to the least-squares solution that maps the estimated trajectory onto the ground truth trajectory.[21]

Similarly, the root mean squared error (RMSE) over all time steps for the translational components can computed by

$$\mathrm{RMSE}(F_{1:n}, \Delta) := (\frac{1}{m}\sum_{i=1}^{m}\|trans(F_i)\|^2)^{\frac{1}{2}}$$

In addition, the collected data includes GPS information, and this can be treated as the reference data for the comparison with the lidar SLAM result.

4.4 Design tools

This section introduces the main design tools used for sensor data collection, lidar data processing, mapping, localization and result visualization.

Robot Operating System (ROS)

Robot Operating System ROS is an open source framework suitable for robot programming. This framework couples the originally loose parts together and provides them with a communication framework. It connects the operating system and the ROS application developed, so it can be regarded as a middleware. In this environment, the robot's perception, decision-making, and control algorithms can be better organized and run.

ROS has following features:

1. Distributed framework

ROS adopts a distributed framework. Through this design, the processes of the robot can be run separately, which is convenient for the modification and customization of modules. Also, this framework improves the fault tolerance of the system.

2. Multi-language support

ROS supports multiple programming languages. C++ and Python are currently the most widely used ROS development languages. In order to support multi-language programming, ROS uses a language-independent interface definition to implement message transmission between modules. In other words, the communication mechanism of ROS has nothing to do with the programming language used. It uses a set of communication interfaces defined by itself.

3. Community

ROS has a huge community that provide debug and visualization tools, packages, and technical supports. There are currently tens of millions of software packages developed using ROS. In addition, ROS complies with the BSD protocol that is completely free for personal and commercial applications and modifications, which also promotes the development of ROS.

In this project, the Rosbag tool are used to record the data from lidars, GPS, IMU and cameras.

<u>Point Cloud Library (PCL)[22]</u>

PCL (The Point Cloud Library) is a large open source C++ library for depth image and point cloud processing. PCL is composed of many advanced algorithms, including filtering, feature estimation, surface reconstruction, registration, model stitching and segmentation. These algorithms have many applications such as filtering outliers in noisy data, combining multiple sets of 3D point clouds, segmenting relevant parts in the scene, extracting key points and calculating geometric shape descriptors for

identifying objects, using point clouds to create and Visualize the surface of objects, etc. PCL has been successfully compiled and configured on Linux, MacOS, Windows, and Android/iOS platforms. To simplify development, PCL is divided into a series of small code libraries that can be compiled separately.
In this project, several filters in PCL are used to process the point cloud data collected in the UWA campus.

Google Cartographer[23]

Google developed a real-time positioning and map building library called cartographer, which developers can use to realize robot positioning and mapping functions in 2D or 3D conditions. Cartographer is designed to obtain relatively high-precision 2D maps in real time when computing resources are limited. Considering that the particle filter method based on simulation strategy has a higher demand for memory and computing resources in a larger environment, cartographer adopts an optimization method based on graph network. Also, it builds a global map based on submap submaps, which can effectively avoid the interference of moving objects in the environment during the mapping process. Cartographer is very suitable for commercial application and redevelopment due to standardized and engineered code. At present cartographer supports multi-sensor data (odometry, IMU and laser scaner etc.) mapping. Google hopes to add more new functions, and support more sensors and robot platforms through subsequent development and community contributions.

KITTI Dataset

The KITTI dataset is jointly founded by Karlsruhe Institute of Technology in Germany and Toyota’s American Institute of Technology. It is currently the world’s largest evaluation dataset for computer vision algorithms in autonomous driving scenarios. This data set is used to evaluate the performance of computer vision technologies such as stereo, optical flow, visual odometry, 3D object detection and 3D tracking in a car environment . KITTI contains real image data collected from urban, rural and highway scenes, with up to 15 vehicles and 30 pedestrians in each image, as well as various degrees of occlusion and truncation. The entire data set consists of 389 pairs of stereoscopic images and optical flow maps, a 39.2 km visual ranging sequence and images of objects labeled with more than 200k 3D, which are sampled and synchronized at a frequency of 10 Hz. In general, the original data set is classified into ‘Road’, ’City’, ’Residential’, ’Campus’ and ’Person’. For 3D object detection, labels are subdivided into car, van, truck, pedestrian, cyclist, tram, and misc.[24]

5. Final design

This section presents the framework of the mapping and localization system, demonstrating thereafter the mapping results for the UWA campus. Ending up with the discussion about the requirement verification.

5.1 Framework

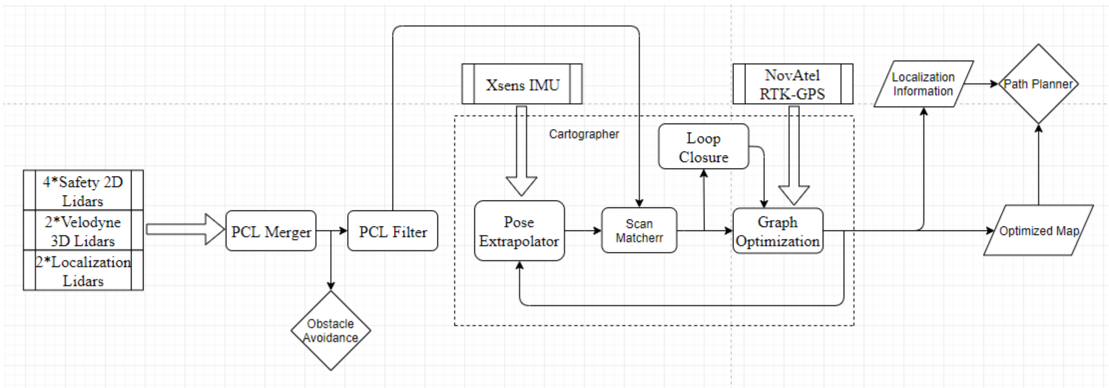

Figure 9. The framework of the mapping and localization system

The mapping and localization system for nUWAy mainly consists of the input module, data preprocessing module and SLAM module.

The input module includes four single-layer lidars, four multi-layer lidars, one Inertial Measurement Unit and one NovAtel GPS. Eight lidars perceive different areas of the environment and provide corresponding point cloud data. Xsens MTi-G-710 IMU can provide angular velocities and accelerations in three dimensions, as well as a filtered GPS signal. NovAtel GPS module is able to work along with a base station and output the RTK-GPS with high precision.

The data preprocessing module is mainly composed of a PCL merger and a PCL filter. The function of this PCL merger is to transform the coordinates of the point sets with respect to respective lidar frame to that with respect to a common base frame, combining then all point clouds within this common frame. This action makes it easy

for the obstacle avoidance module and the SLAM module to process the point cloud.
The PCL merger is followed by a PCL filter which can be divided into two components, a bandpass filter and a voxel filter. The bandpass filter serves to filter out the points outside a certain min and max range, because the lidar which is partially covered by the vehicle may generate some hit points close to the it, and some far hit points may be noisy and irrelevant to the mapping process. The voxel filter is responsible for "downsampling raw points into cubes of a constant size and only keeping the centroid of each cube."[14] This action is able to reduce the density of the original point cloud while keeping the geometric features of it, so that the computational load can significantly decrease. One example of the PCL filter for nUWAy shuttle bus is shown in Figure 10.

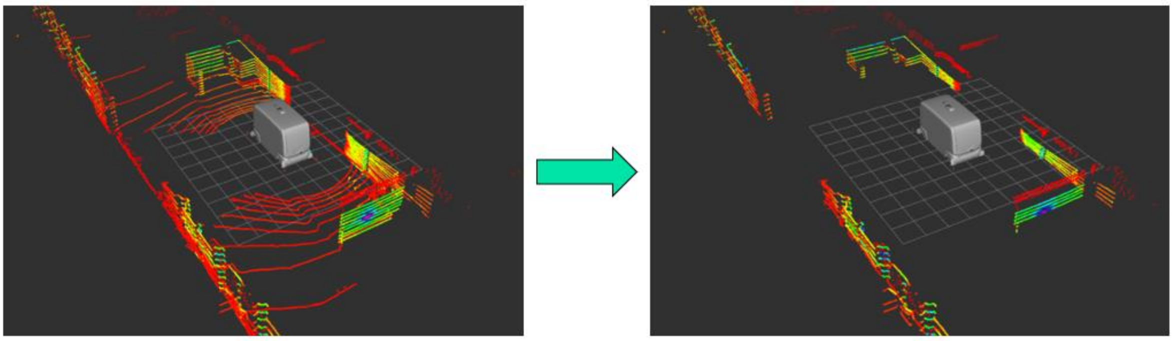

Figure 10. An example of the PCL filter

The SLAM module is implemented "Cartographer_ros" package developed by Google. This module consists of the pose extrapolator, the scan matcher, loop closure and backend optimization.
The pose extrapolator fuses the previous estimated pose from the scan matcher and the IMU data based on Unscented Kalman Filter in order to update the pose. This updated pose will be regarded as a decent initial estimate for the scan matcher.[25]
The scan matcher serves to estimate the pose of the autonomous car based on the scan-to-map method. Once the optimal pose is found, the probabilities of grids in the submap will be updated accordingly. In the pose estimation, the scan matcher tries to solve a nonlinear least squares problem as follow:

$$\underset{\xi}{\operatorname{argmin}} \quad \sum_{k=1}^{K} \left(1 - M_{\text{smooth}}(T_\xi h_k)\right)^2$$

Where T represents a transform between the scan frame and the submap frame, h_k is the k_{th} scan point. M is a function mapping coordinates to probability values.[12]

After the optimal pose of a scan is determined, the probability of the submap grid x is updated as below:

$$\text{odds}(p) = \frac{p}{1-p},$$

$$M_{\text{new}}(x) = \text{clamp}(\text{odds}^{-1}(\text{odds}(M_{\text{old}}(x)) \cdot \text{odds}(p_{\text{hit}})))$$

Where p_{hit} is the initial probability for the grid which contains the hit points.[12]

The created submaps would be stored in the memory for loop closure and the backend graph optimization. When the current estimated pose is close to a visited submap, a scan matcher tries to find the corresponding observation in a search window around the submap. In this step, a variant branch-and-bound approach is used to speed up loop closure process. Once a good solution is found, it is added as a loop closing constraint to the sparse pose adjustment (SPA) problem.[10] Also, as shown in the Figure 9, GPS is also added as a constraint to the global optimization problem. The graph optimization process is able to "re-arrange submaps between each other so that they form a coherent global map by solving this global optimization problem."[14]

Finally, the offline generated map along with the real-time localization information feed into the path planner for the navigation task.

5.2 Results

Two maps generated from the presented framework are listed below, including the main campus, and the place outside our REV lab.

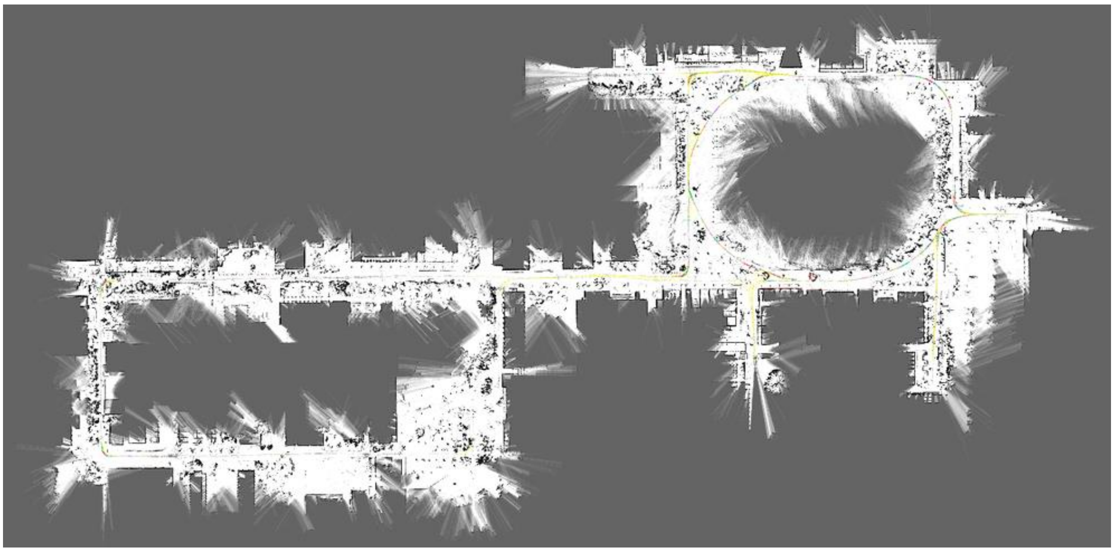

Figure 11. The map of UWA campus

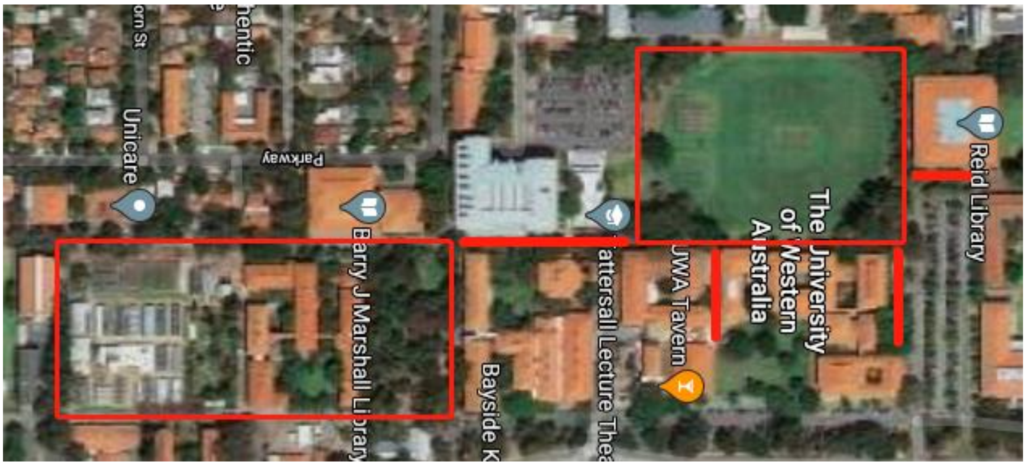

Figure 12. Google map for UWA campus

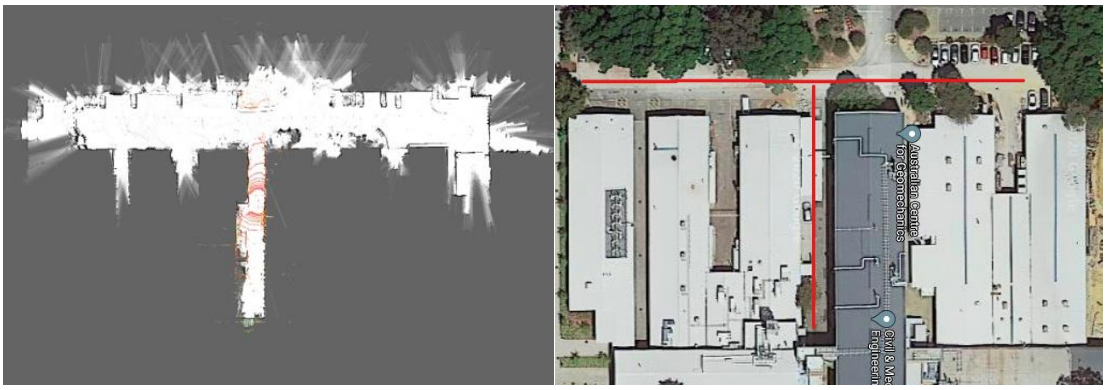

Figure 13. Maps outside the REV lab

5.3 Discussion

This section discusses the traceability of the final design and several aspects which should be paid attention to.

The map resolution is an important parameter which affects the performance of the autonomy system. It is a trade off between the localization accuracy and running time for the localizer and the path planner. To be specific, if the localizer uses a map with the high resolution (the pixel dimension is lower than 10cm), the estimated pose would be more accurate since the detailed information of the environment is recorded in the map. However, It takes more time for both the localizer and the path planner to give the results, since their running time both depends on the number of the pixels. If the resolution is low (the pixel dimension is larger than 50cm), the localization accuracy and the drivable area drops although the computational load decreases, as shown in Figure 14. By experiments, an appropriate pixel dimension is around 20cm, bringing in the acceptable positioning accuracy and running time.

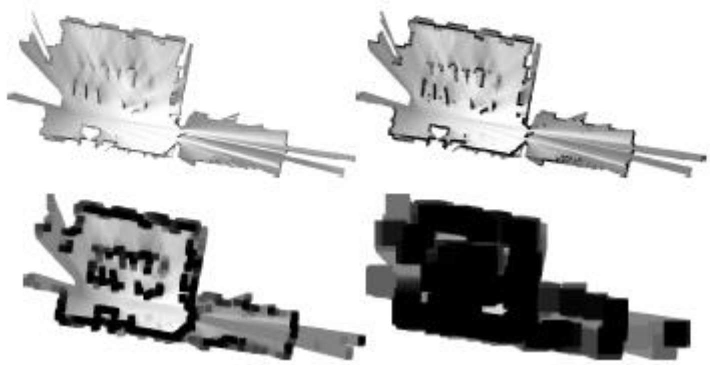

Figure 14. The same map with different resolution[12]

Also, the presented framework has the capability resistant to the effect of the moving objects in the environment. The map takes the form of probability grids, which means each grid in the map accommodates a changeable probability value to represent the odds of being occupied[12]. During the mapping process, the system continuously perceives the environment and updates the probability values in the grids. The probability of occupancy of the cell increases only if a moving object passes through it, and then the value will be updated to a lower value.

In addition, four requirements mentioned before are partially satisfied.

1. The PCL merger is able to combine point clouds from eight lidars.
2. The SLAM framework contains the loop closure and global optimization process, which can handle the accumulated error. The mapping result also demonstrates the drift issue is addressed in the large scene given the proposed framework.
3. Multiple sensors such as IMU, GPS and Lidars can be fused in the framework. When one sensor is restricted for some reasons, the other can complement it. In this project, the lidar mapping ran into issues when the shuttle bus was moving on route across James Oval. The mapping result demonstrates this problem is fixed if adding the IMU into the framework. However, this requirement is not fully satisfied. The GPS for now cannot be integrated in cartographer for 2D mode. The slope information cannot be recorded in the 2D map neither.
4. Three aspects contribute to the real-time localization. A variant branch-and-bound approach is used to speed up loop closure process. Also, an acceptable map resolution and the PCL voxel filter aid to reduce the computational load.

6. Future work

According to above contents, the requirements are not fully satisfied. Firstly, in dynamic environment the localizer may be unable to give the estimated pose with sufficient accuracy due to the differences between the environment and the static map. In this case, The RTK-GPS can be considered to be added into the SLAM framework. The high-precision GPS signal can be treated as a decent initial pose to be further processed, and it can significantly reduce the drift. Secondly, the present map takes the form of 2D grids in which the slope information cannot be recorded. This would heavily impact the localization accuracy if there are many slopes in the environment. Therefore, creating 3D map is a topic for further investigation.

Sematic SLAM is another topic which is worthy to research. Adding sematic information can benefit to both localization and mapping. For example, dynamic objects can be identified according to semantic information, and then removed from the map. Objects with semantic tags can be associated with data, which then works in conjunction with geometric information to improve the mapping accuracy.[26]

7. Conclusion

In conclusion, this thesis firstly introduces the project background and scope, and then focuses on the existing issues under this background. Four main problems are identified in the nUWAy project, which are relevant to the drift issue, computational load, sensor fusion and point cloud merging. After that, the author further investigates the recent work related to Lidar-based SLAM, focusing on the presented approaches, achieved outcomes and how these can be used for problems in the nUWAy project. From researching, cartographer SLAM developed by Google holds promise for reaching the goal of this research project. According to the design requirements, a SLAM framework based on Google cartographer is presented. The mapping result for the UWA main campus and further discussion related to the map resolution and moving objects demonstrates this expanded approach has its own advantages against the problems identified in this specific scenario.

8. Reference

[1] Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., ... & Veronese, L. (2019). Self-driving cars: A survey. *arXiv preprint arXiv:1901.04407*.

[2] Khairuddin, A. R., Talib, M. S., & Haron, H. (2015, November). Review on simultaneous localization and mapping (SLAM). In *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)* (pp. 85-90). IEEE.

[3] Feder, H., Leonard, J., & Smith, C. (1999). Adaptive Mobile Robot Navigation and Mapping. *The International Journal Of Robotics Research*, *18*(7), 650-668. doi:10.1177/02783649922066484

[4] Leonard, J., & Durrant-Whyte, H. (1992). *Directed sonar sensing for mobile robot navigation*. Boston: Kluwer Academic Publishers.

[5] Leonard, J., & Feder, H. (2001). Decoupled stochastic mapping [for mobile robot & AUV navigation]. *IEEE Journal Of Oceanic Engineering*, *26*(4), 561-571. doi: 10.1109/48.972094

[6] EasyMile. (2015). *EasyMile EZ10 User Manual* [Ebook].

[7] Chong, T., Tang, X., Leng, C., Yogeswaran, M., Ng, O., & Chong, Y. (2015). Sensor Technologies and Simultaneous Localization and Mapping (SLAM). *Procedia Computer Science*, *76*, 174-179. doi: 10.1016/j.procs.2015.12.336

[8] Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Transactions On Intelligent Vehicles*, *2*(3), 194-220. doi: 10.1109/tiv.2017.2749181

[9] Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions On Robotics*, *23*(1), 34-46. doi: 10.1109/tro.2006.889486

[10] Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., & Vincent, R. (2010, October). Efficient sparse pose adjustment for 2D mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 22-29). IEEE.

[11] Kohlbrecher, S., Von Stryk, O., Meyer, J., & Klingauf, U. (2011, November). A flexible and scalable slam system with full 3d motion estimation. In *2011 IEEE*

International Symposium on Safety, Security, and Rescue Robotics (pp. 155-160). IEEE.

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

[13]Zhang J, Singh S. LOAM: lidar odometry and mapping in real-time [C/OL]/ Proc of Robotics: Science and Systems. (2014-07-12)[2018-08-12]. http://www.roboticsproceedings. org/rss10/p07. pdf.

[14] Algorithm walkthrough for tuning — Cartographer ROS documentation. (2020). Retrieved 3 May 2020, from https://google-cartographer-ros.readthedocs.io/en/latest/algo_walkthrough.html

[15] robot_localization wiki — robot_localization 2.5.6 documentation. (2020). Retrieved 3 May 2020, from http://docs.ros.org/lunar/api/robot_localization/html/index.html

[16] The REV Project. (2020). Retrieved 3 May 2020, from https://therevproject.com/vehicles/nuway.php

[17] Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, *13*(2), 99-110. doi: 10.1109/mra.2006.1638022

[18] Stahl, T., Wischnewski, A., Betz, J., & Lienkamp, M. (2019, January). ROS-based localization of a race vehicle at high-speed using LIDAR. In *E3S Web of Conferences* (Vol. 95, p. 04002).

[19] Jaimez, M., Monroy, J. G., & Gonzalez-Jimenez, J. (2016, May). Planar odometry from a radial laser scanner. A range flow-based approach. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4479-4485). IEEE.

[20] Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012, October). A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 573-580). IEEE.

[21] Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *Josa a*, *4*(4), 629-642.

[22] Library, P. (2020). *Point Cloud Library*. Point Cloud Library. Retrieved 19 December 2020, from https://pointclouds.org/.

[23] *Cartographer ROS Integration — Cartographer ROS documentation*. Google-cartographer-ros.readthedocs.io. (2020). Retrieved 19 December 2020, from https://google-cartographer-ros.readthedocs.io/en/latest/index.html.

[24] *KITTI data set data first experience - Programmer Sought*. Programmersought.com. (2020). Retrieved 19 December 2020, from https://www.programmersought.com/article/86624345131/.

[25] Olson, E. B. (2009, May). Real-time correlative scan matching. In *2009 IEEE International Conference on Robotics and Automation* (pp. 4387-4393). IEEE.

[26] Chen, X., Milioto, A., Palazzolo, E., Giguère, P., Behley, J., & Stachniss, C. (2019, November). Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4530-4537). IEEE.

[27] Censi, A. (2008, May). An ICP variant using a point-to-line metric. In *2008 IEEE International Conference on Robotics and Automation* (pp. 19-25). Ieee.