博士論文

Doctoral Dissertation

Evaluation and Quantification of Digital Map Capability for Vehicle Self-Localization

（自車位置推定のためのディジタル地図性能の評価と定量化に関する研究）

Ehsan Javanmardi

ジャワンマーディ エッサン

Abstract

The autonomous vehicles (AV) have gained a focus as an essential technology within intelligent transportation systems since their introduction in the early 1950s. Since then, many attempts have been made to bring this intelligent vehicle on the roads fully or partially to benefit from its capabilities. In order to make this technology work in the real-world conditions as well as the controlled environment in laboratories, researchers have encountered many challenges. One of these challenges is accurate vehicle self-localization.

Accurate and robust self-localization is an essential task for AVs. On the other hand, high precision self-localization solution when combined with a prebuilt map can simplify the difficult concept of perception and scene-understanding into a less complex positioning problem. Conventional approaches use global navigation satellite system (GNSS) for autonomous vehicle self-localization. This technology is low cost and works very well in the open sky. However, in the urban environment, the accuracy degrades dramatically due to non-line of sight (NLOS) and multipath reception of the satellites' signals and make it difficult for an AV application.

In recent years, use of light detection and ranging (LiDAR) for the perception of AVs become more popular due to its price down, miniaturization and density enhancement. Compared to cameras, LiDARs are more reliable because problems caused by illumination change, light conditions and shadows do not affect them. On the other hand, LiDARs can obtain more accurate distance information comparing to the stereo cameras which make them more suitable for the self-localization applications. LiDAR-based localization can be divided into two main categories. Map-based and without a map which is also known as simultaneous localization and mapping (SLAM). In the SLAM methods, there is no prebuilt map. Based on the current position, the surrounding information is stored online as a map. In next time stamp, this stored map (information) are used to calculate the displacement of the vehicle. According to this displacement, the current position of the vehicle is calculated, and again the surrounding information is stored as a map to be used of next time stamp. SLAM methods work well over a short distance; however, due to its dependent nature, they are still suffering from accumulative error in long distances.

Therefore, in recent years, map-based methods have gained more attention in most of the AV platforms. In a map-based method, a raw point cloud of the environment is

collected offline using high-end mobile mapping systems (MMSs). Then, based on the map formats, the map is generated from raw point cloud data. Later, in the self-localization phase, the scan acquired from LiDAR mounted on the top of the vehicle is matched to the map to obtain the position of the vehicle within the map. In map-matching based self-localization, as each sequence are independent, there is no error accumulation.

In this dissertation, the focus is made on the map based self-localization methods and several contributions have been made.

In the map based categories, map plays a significant role in achieving high accuracy self-localization. For accurate self-localization, the global and local accuracy of the map is essential as well, and many types of research have been done to obtain such a highly accurate map. Strategic Innovation Promotion Program (SIP) has defined the required global accuracy of map for autonomous driving to be less than 25cm. This requirement comes from the satellite image resolution and vehicle's tier width. There is a misunderstanding that if the map is accurate, then the localization within the map will be accurate as well. In fact the highly accurate map does not guarantee the accuracy of the localization. In other words, map accuracy is different than the ability of the map for localization. For example, in the case of the tunnel, no matter how much the map is locally and globally accurate, the lack of longitudinal features in the map causes localization error in the moving direction. To achieve accurate self-localization within a map, the map should satisfy some requirements. In other words, the map should meet some specific criteria which define the ability of the map for self-localization. To the best of this author's knowledge, there is no comprehensive study of the definition and formulation of these criteria. Therefore in this dissertation, for the first time, the required criteria regarding the ability of the map for accurate self-localization are defined and formulated.

Some of these criteria highly related to the environment and as surrounding environments are different from place to place in the map, it should be evaluated by defined criteria. And some other criteria are related to the quality of representation of the environment by the map. In addition to the sensor related parameters in the mapping phase such as frequency of laser scanner, layer count, the range of the beam, setup parameters, etc., quality of representation highly related to the format and abstraction ratio (resolution) of the map. Quality of representation of the environment in each of the map formats is different. In other words, some map formats discard more details of the map comparing to the others. This information loss of the map might change the quality of some of the criteria and lead to a localization error. However, in some part of the map, abstraction does not necessarily change the quality of the map, or rate of change is

acceptable. Additionally, in some cases, other criteria might compensate the lack in one criterion. Therefore, in order to evaluate the self-localization ability of the map at a specific point, all criteria should be considered together.

In this dissertation, four general criteria for the map are defined. These criteria are feature sufficiency, layout, representation quality, and local similarity of the map. These criteria are defined regardless of the map format and can be applied to any other map formats. However, in this work, to quantify each of these criteria, the focus is made on the ND map format, and several factors are defined. For each point in the map, these factors are calculated based on the features in surroundings called local vicinity. By obtaining the correlation of the map factors with localization error, the effectiveness of the factors is investigated. Additionally, by applying principal component regression (PCR), the predictability of the self-localization error based on these factors are investigated. To evaluate the predictability of the defined factors, experiments have been conducted in Shinjuku, Tokyo, Japan. The route of experiments is around 40Km. The experimental results show the error modeled from the factors can represent the localization error of the map in 71% of cases with an error lower than 10cm. In order to increase the accuracy of prediction, the factors are fed to a simple feedforward neural network to model the error. The result is improved and around 78% of the localization error can be modeled with lower than 10cm accuracy.

The outcome of the proposed map evaluation framework can be used for evaluating the map before its use for localization. Also the results of this study can be used for evaluation of the map which is not made specifically for the self-localization purpose.

As the other contribution of this dissertation, the outcome of the map evaluation framework is used for adaptive determination of the map and localization parameters. These parameters are map resolution and range of the laser scanner. Basically lower map resolution is more desirable as it can save the computation time and memory size, but it might remove some important information for localization. In order to determine the best resolution of the map, the map is evaluated for different resolution using map evaluation factors and the optimum resolution is selected. On the other hand, the range of the laser scanner is an important parameter for reducing the matching time. This parameter can be determined adaptively based on the quality and quantity of the features in the surroundings and stored in the map beforehand. In this dissertation, the adaptive determination of the laser range is proposed which uses the map evaluation factors to define the optimum laser range for different part of the map.

In addition to the aforementioned contributions, globally accurate urban mobile

mapping frameworks are proposed. One of the challenges of current Mobile Mapping Systems (MMS) is their precision, accuracy and the cost. In order to get required accuracy in the urban area, current MMSs, first converge their position system in non-urban area which takes more than 10 minutes and then traverse through the urban area. And after few kilometers the positioning system needs to be re-converged in non-urban area. This process is so much labor and time intensive. Even with this, the generated map has around a meter of error. Therefore in order to remove this error post-calibration with labor-intensive Global Control Point (GCP) is performed which collection of GCP and post-calibration task costs around 10 million yen for each square kilometers for the companies.

In order to remove this cost, in the proposed urban mapping frameworks, two laser scanner is used in addition to the Inertial measurement unit (IMU), odometer, and GPS. One laser scanner is used to perform the simultaneous localization and mapping and the other is tilted downward to capture the environments as point cloud map. The airborne image is used as a reference to correct the trajectory of the vehicle. In order to correct the trajectory of the mapping vehicle, the lane markings extracted from tilted laser scanner is registered to the lane marking extracted from the airborne image. The map which is used for this dissertation is all made by this mapping system and global accuracy of the map is very high.

In this dissertation, two self-localization methods based on abstract map format which are multilayered vector map and the probabilistic planar surface map are proposed as well. One of the huge disadvantage of the map based methods are map size. For a small area, for instance 300m to 300m, around 250 million points should be stored to the autonomous vehicle storage system. On the other hand, the matching algorithm requires high computational time to deal with this data size. In the multilayered 2D vector map, different layers of building footprints are stored as a vector in the map. By using different layers of the building, the feature for matching increases and make the matching more accurate. For each vector, the variance of the points are stored in the map to allow the localization process to rely on more certain vectors rather than an uncertain one. This can reduce the map size to several thousand times. In the probabilistic planar surface map, instead of using heavy point cloud, the planar surfaces are extracted from the point cloud map. Each planar surface contains the variance of the points which shows the uncertainty of the planes. Storing the variance for each plane help the localization process to rely more on the certain walls. In the matching phase, distortion of the input scan is one of the challenges which makes the misalignment of the input scan to the map. In this work, distortion is removed in the optimization process. In fact, in each iteration of the

optimization, the scan is reshaped by the new results and the distortion is removed to fit better to the map.

This dissertation also includes an overview of the state of the art localization methods, and also a discussion about the different source of the error in self-localization based on map-matching.

To my dear family

تقدیم به خانواده عزیزم

Acknowledgment

These sentences are my deepest gratitude to all people who supported and encouraged me during my doctoral research in the University of Tokyo, Japan.

First and foremost, I would like to express my biggest appreciation to my supervisor, Prof. Shunsuke Kamijo, for giving me the opportunity to be a member of the Kamijo Laboratory, and pursuing my PhD. I am sincerely grateful to his moral support, steady encouragement, extraordinarily tolerant, and insightful advice. I had a great time at the Kamijo lab, and I hope our collaborations will continue in the future.

I would like to give special thanks to Prof. Wei-Bin Zhang for the opportunity of the half-year research visit at the California PATH Program of the University of California, Berkeley, and all the support he has shown me throughout my research and career.

I am grateful to my Dissertation Committee Prof. Asami, Prof. Aizawa, Prof. Naemura, and Prof. Yamasaki, for providing me exensive personal and professional guidance and teaching me a great deal about both scientific research and life.

My special thanks go to my dear brother and colleague Mahdi Javanmardi who taught me a lot of novel ideas in the research. I always admired his creativity and hard work. In addition to academic research, his advice for my various difficulties was incredibly helpful.

I would like to thank Dr. Yanlei Gu for his continuous encouragement, constructive supports during my graduate study. The discussions with him were always enjoyable. I am also thankful to our secretary Ms. Miwa for her endless support which I have greatly benefited.

I gratefully acknowledge the funding received towards my PhD from the Ministry of Education, Culture, Sports, Science and Technology (MEXT). I also want to thank the Graduate Program for Social ICT Global Creative Leaders (GCL) of The University of Tokyo by Japan society for the promotion of science (JSPS) for supporting my research.

Finally but not least, I would like to express my gratitude to my parents and younger brother Sadjad for their moral support, constant understanding, and sincere encouragement.

Ehsan Javanmardi

ジャワンマーディ　エッサン

Table of contents

Abstract ii

Acknowledgment viii

Table of contents x

List of Figures xii

Chapter 1. Introduction 1

Chapter 2. Vehicle self-localization and digital map 8

Chapter 3. Low-cost urban mapping framework 16

3.1. Introduction 16

3.2. Mapping Platform 17

3.3. Mapping Framework 19

3.4. Updating vehicle trajectory utilizing Aerial Data 20

Chapter 4. Map abstraction for Self-localization 27

4.1. Introduction 27

4.2. Vehicle Self-Localization Based on Multilayer 2D Vector Map 30

4.2.1. Generating Multilayer Vector Map 30

4.2.2. Scan-to-Vector-Map Matching for Localization 33

4.3. Vehicle Self-Localization Based on Probabilistic Planar Surface Map 37

4.3.1. Generating Probabilistic Planar Surface Map 38

4.3.2. Scan-To- Planar -Map Matching for Localization 41

4.4. Experimental Results 44

4.4.1. Evaluation for Multiplayer Vector Map-based Localization 45

4.4.2. Evaluation for Plane Surface Map based Localization 47

Chapter 5. Quantification of digital map ability for self-localization 50

5.1. Introduction 50

5.2. Evaluation criteria 56

5.2.1. Feature sufficiency of the map 56

5.2.2. Layout of the map 58

5.2.3. Local similarity of the map 60

5.2.4. Representation quality of the map 60

5.2.5. Formulation of the evaluation criteria 62

5.3. Factors for feature sufficiency 63

5.3.1. $\boldsymbol{feature\_count}$, $\boldsymbol{Dm\_count}$, and $\boldsymbol{Dm\_ratio}$ 63

5.3.2. ***occupoancy_ratio*** 67

5.3.3. ***long_weight*** and ***lat_weight*** 69

5.4. Factors for the layout 70

5.4.1. ***Feature DOP*** 70

5.4.2. ***normal_entropy*** 71

5.4.3. ***angular_entropy*** 73

5.4.4. ***r_average*** 74

5.5. Factor for representation quality 74

5.6. Factors for local similarity 75

5.6.1. ***score_entropy*** 75

5.6.2. ***pfh_similairty*** 77

5.6.3. ***D_Battacharyya*** 78

Chapter 6. Digital map evaluation based on quantification factors 79

6.1. Experimental methodology and setup 79

6.2. Evaluation and error modeling with PCR 81

6.3. Error modeling using neural network 90

Chapter 7. Adaptive refinement of map and localization parameters based on map factors 93

7.1. Adaptive refinement of map resolution 93

7.1.1. Effect of map resolution to localization error 94

7.1.2. Effect of resolution in NDT-based localization 97

7.1.3. Map refinement based on factors 100

7.1.4. Experimental results 103

7.2. Adaptive determination of the laser range 106

7.2.1. Effect of laser range in self-localization 106

7.2.2. Adaptive determination of laser range 110

7.2.3. Experimental results 111

Chapter 8. Conclusion 114

Appendix GCL Social Innovation Project Towards Autonomous Vehicle Platform for Smart Society 116

Overview 116

Social Innovation Project 117

Conclusion 139

GCL Acknowledgment 140

List of publications 141

References 143

List of Figures

Figure 1-1 Assist of self-localization for various application such as augmented reality (AR)...........1
Figure 1-2 GCP extracted for 1km^2 to post calibrate the MMS scan..3
Figure 1-3 Overview of abstract map based localization techniques. Left image shows the multilayerded 2D vector map method. The right image shows the probabilistic planar surface map based localization...4
Figure 1-4 **.** Overview of the proposed map evaluation method...6
Figure 1-5 Different map resolution obtained for different part of the map by adaptive refinement method 7
Figure 2-1 GPS NLOS detection. (a) Using dual polarization technique the NLOS signal is detected [31]. (b) Using Sky-pointing camera the occluded satellite are detected [33]......................................9
Figure 2-2 Velodyne laser scanner for different autonomous vehicle platform. (a) from left, a 64 layer Velodyne laser scanner (HDL-64), 32 layer (32-DL) and 16 layer (VLP-16). (b) HDL-64 is mounted on Uber autonomous vehicle. (c) VLP-16 is mounted on Fords autonomous vehicle........................10
Figure 2-3 (a) Velodyne HDL-64 scan data which is captured in California, U.S. (b) Data from Luminar laser scanner. Luminar laser scanner does not rotate and it is stationary. 11
Figure 2-4 Lidar based self-localization categories. SLAM, Map matching......................................13
Figure 2-5 Methods which is used laser scanner and prior map for self-localization [29] [33].........14
Figure 2-6 Point cloud map of the Hitotsubashi, Tokyo, Japan. This point cloud is the combination of the (Mobile Mapping System) MMS data and airborne laser scanner (ALS) data.............................15
Figure 3-1. Generalized MMS workflow [13]..16
Figure 3-2. MMS which is used in part of mapping of the experimental area...................................17
Figure 3-3. Proposed mapping system...18
Figure 3-4. Equipment which is used for mapping..18
Figure 3-5. Equipment which is used for mapping..19
Figure 3-6. Calibration of aerial image using few GCPs...20
Figure 3-7. GCP acquisition in urban areas, a labor-intensive and time-consuming task..................20
Figure 3-8. Landmark updating technique for calibrating 3D point cloud data obtained from ground surveillance system in the urban areas...21
Figure 3-9. Framework of the filter including the aerial image data for the calibration....................22
Figure 3-10. Generation of the normal distribution from the reference data: (a) Estimated 1D normal distribution of the sample points; (b) Estimated 2D normal distribution of the sample points; (c) Extracted aerial road markings; (b) Generated NDT map with 2m grid size......................................23
Figure 3-11. The overall workflow of the proposed 2D MMS registration framework......................25
Figure 3-12. Generated map using the proposed mapping frameworks. The point cloud are subsampled

by 10cm grids..26

Figure 4-1. The concept of the vehicle self-localization using the proposed abstract map formats rather than massive 3D point cloud..31

Figure 4-2. Building footprints for different layers (height *Z*) of the building. 31

Figure 4-3. The procedure of generating the multilayer 2D vector map from the 3D point cloud. (a) Top view of the original 3D point cloud. (b) The building points extracted from the point cloud. (c) Multilayer 2D vector map. (d) Normal distributions of the vector map. 32

Figure 4-4. Calculating the vector uncertainty. The thick navy blue line shows the vector representing a building wall. The red box shows the bounding box to get uncertainty of vector representing the wall, and the green vectors are the Eigen vectors $u1, u2$ used for generating uncertainty for the vector. The green ellipse shows the normal distribution generated based on the center point of the vector and its uncertainty...33

Figure 4-5. Flowchart of the vector map-based self-localization. Inputs are the vector map and point cloud acquired by the laser scanner, and the output is the current position of the vehicle. 34

Figure 4-6. Normal Distributions generated from the vectors. 35

Figure 4-7. A few examples of the localization technique based on the multi-layer 2D vector map. The input laser scans are shown in red. The blue spheres are the NDs generated from the vector map. ...38

Figure 4-8. Flowchart of the planar surface map-based vehicle self-localization. Inputs: the planar surface map and 3D laser scan. Output: the 6D state of the vehicle. 39

Figure 4-9. The procedure of generating the planar surface map. (a) The original 3D point cloud. (b) The ground segmentation result. (c) The extracted ground. (d) The Gaussian mixture model of the ground surface. (e) The off-ground points. (f) The planar surface map of the off-ground points.......40

Figure 4-10. Comparison of different map formats and abstraction levels: point cloud map, fixed-size grid ND map, and 3D planar surface map. 41

Figure 4-11. Effect of the NDs generated from the planar surface map on the defining the correct corresponding ND of the points in the input scan. (a) Misdetection of the nearest ND due to the large ND domain, and the score reduction due to the distance of the input points from the center of the ND. (b) Effect of representing the planar surface using smaller NDs. 43

Figure 4-12. The experimental area in the vicinity of Hitotsubashi, a typical urban area in the Chiyoda-ku area of central Tokyo, Japan. The red line shows the experiment route and the green dots show the positon acquired by GPS for the route. 45

Figure 4-13. The experimental setup. The left is the MMS used for the mapping and the right is our experimental vehicle. Velodyne VLP-16 is mounted on the top of this car at a height of 2.3m. 45

Figure 4-14. First chart of (a) shows the evaluation of proposed method by comparing to ground truth, and as result shows, the mean error is less than 20 cm. Thus, the proposed method extremely decreased the map size while preserving the localization accuracy. Other chart of (a) show the comparison of proposed method with 2D point cloud method. Figure (b) shows the number of map elements in each map structure..46

Figure 4-15. Comparison of proposed method with 2D point cloud with different grid size. Height of building footprint used in 2D point cloud methods is 6m. As results show proposed method perform better comparing to all grid sizes. 47

Figure 4-16. Comparison between proposed method and 2D point cloud which uses all footprints in different heights. 48

Figure 4-17. Map size comparison of proposed plane surface map and other point cloud based methods. A base-10 log scale is used for the Y axis. As can be seen in this figure, planar surface map structure is extremely lighter than others...48

Figure 4-18. Accuracy of the plane surface based self-localization and other NDT-based methods. .49

Figure 5-1 . Overview of the proposed map evaluation method. For a sample point P (top left), the map factors in the local vicinity are calculated (right). Based on these factors, the localization error is estimated (bottom left)...53

Figure 5-2 . Different layout for the map. In (A) all features are lean to one corner. This layout can cause positioning error. In (B), the distribution of the features are even. 57

Figure 5-3 . Different layout for the map. In (A) all features are lean to one corner. This layout can cause positioning error. In (B), the distribution of the features are even. 58

Figure 5-4 Effects of satellite distribution to the GDOP. 59

Figure 5-5 Different situations (A and B) in that the localization in longitudinal direction has error.59

Figure 5-6 Situation that the map does not have any lateral feature to get the longitudinal position. 60

Figure 5-7 Local similarity cause the positioning uncertainty. Longitudinal position cannot be obtained. The blue shapes are the buildings which surround the laser scanner. The red lines are scan on the specific time and shows the observed part of the building by laser scanner. In this scenario, vehicle cannot distinguish between position A and B. 61

Figure 5-8 Shift of score peak due to abstraction of the map. Top images shows the 2.0m ND map and the matching score of scan (red points) on it. Score of 2.0m ND map does not have any shift. Bottom image shows the same pair for 4.0m ND map. In 4.0m ND map, due to higher abstraction of the map, peak of matching score is shifted by 60cm. This shift of peak at least cause 60cm localization error. 62

Figure 5-9 Local vicinity of target point P. Local vicinity is extracted based on the range and vertical angle ψ of the laser scanner. (a) ND map of local vicinity. (b) point cloud map is the local vicinity. 64

Figure 5-10 Definition of 1D, 2D and 3D features. 1D feature can perform the positioning for 2 dimension. 2D features can perform in 1 dimension and 3D object has high uncertainty so it cannot get the correct location in any direction...65

Figure 5-11 feature dimension value for three types of objects. 65

Figure 5-12 1D, 2D and 3D shapes in point cloud map. The red features are 1D features and the blue features are 2D features and the green s 3D features. 66

Figure 5-13 1D (red), 2D(blue) and 3D (green) features in map. Left image is point cloud amp and right image is the ND map generated from left point cloud with 2.0m grid. Ground and building walls are 2D features and poles and trunk of trees are 1D and vegetation and trees has 3D features. 67

Figure 5-14 Depth images (bottom images) correspond to the features in local vicinity (purple ellipsoids). In depth image white shows presence of features and black shows the absence of features. Occupancy ratio for the left image is 0.354 and for the left is 0.523. In right scenario, local vicinity is occupied with more features thus the occupancy ratio is higher. 68

Figure 5-15 Quality of localization in longitudinal and lateral is depend on the direction of feature’s normal. Lateral score and longitudinal score is shown with purple. Uncertainty in lateral is lower, thus lateral weight should be higher than longitudinal weight. Blue ellipsoid shows the ND_map and red points is input scan. 68

Figure 5-16 The distance to features for different map format. 71

Figure 5-17 Comparison of the layout (distribution) of the feature with lateral and longitudinal FDOP. 71

Figure 5-18 Vehicle position uncertainty (red ellipsoid) in different layout of the building. (a) has high uncertainty in moving direction. (b) has high localization error in lateral direction. (c) layout of the building make the localization more accurate. 72

Figure 5-19 Effect of map resolution on the normal of the features. Red arrows are the normal of each featue (ND). 73

Figure 5-20 Different Mahalanobis distance values for different NDs. Grid size is 2.0m. Blue shows lower mahalanobis distance and red shows higher values. Ground shows low value means the representation quality is high and vegetation and trees have higher Mahalanobis value means the representation quality is low. 75

Figure 5-21 Score for two different scenario. Left environment has local similarity and right does not. Left score has more peaks. Score is obtained by moving point cloud (green points) over ND map (blue ellipsoid) and calculating likelihood. 77

Figure 6-1 path of experiments in Shinjuku. 80

Figure 6-2 Our experimental vehicle. Front Velodyne VLP-16 is used for mapping and it is tilted to scan the environment densely (pith -80°). Top VLP-16 used for localization. 80

Figure 6-3 sample points and particles which are distributed in the vicinity of the sample point. 81

Figure 6-4 Comparison of different factors with 3D maximum error. For each figure, blue graph is 3D maximum error and orange graph is values of factors. Normal entropy factor is calculated for 8×8 bins. 82

Figure 6-5 Factors related to the feature sufficiency for path II (2.0m ND) compared to mean and max error of the same path. 83

Figure 6-6 Path #I, #II, and #VI of the experimental area. 83

Figure 6-7 Factors related to the layout for path II (2.0m ND) compared to mean and max error of the same path. 84

Figure 6-8 Factors related to the representation quality and local similarity for path II (2.0m ND) compared to mean and max error of the same path. 85

Figure 6-9 Modeled error with PCR for Path #I with (bottom) and without similarity (top). 87

Figure 6-10 Modeled error with PCR for Path #II with (bottom) and without similarity (top). 88

Figure 6-11 Modeled error with PCR for Path #VI with (bottom) and without similarity (top). 89

Figure 6-12 Modeled error (green) vs actual error (blue) for the path #I, #II, #VI. 91

Figure 7-1 Different resolution of the point cloud. Right image is the high density with original resolution. The left image is down-sampled by 30cm grids. 94

Figure 7-2 Different resolution for ND map. For the ND map, grid size can adjust the resolution. 95

Figure 7-3 Different resolution for the vector map. The vector map resolution can be adjusted using line extraction parameters. 95

Figure 7-4 Different resolution of plane extraction algorithm. If we set the parameter of plane extraction to very high value, then we have very coarse plane as shown in the left. 96

Figure 7-5 Special case that the vehicle self-localization longitudinal accuracy is low (a). In (b) the resolution of the map is good enough so we can see the longitudinal features and we can obtain the longitudinal position as well as lateral position. In (c) by adding a new wall in front, the vehicle can obtain the longitudinal position as well. 97

Figure 7-6 Localization error for different map resolution. Red region shows errors more than 0.25m. 98

Figure 7-7 Path of experiments in Shinjuku, Tokyo (left), and its point cloud map (right). 100

Figure 7-8 Vehicle position uncertainty in different scenarios. Red ellipsoid means high error and green ellipsoids means low position uncertainty. 101

Figure 7-9 Modeled error using map evaluation factors for different grid sizes (2.0m~5.0m) 101

Figure 7-10 Best grid size of the map before (blue) and after (red) final process. 103

Figure 7-11 Best grid size of the map for experimental path before (blue) and after (red) final process for 0.25m error threshold. 103

Figure 7-12 Best grid size of the map for experimental path before (blue) and after (red) final process. 104

Figure 7-13 Estimated best grid size for the experimental path. Top figure is for 0.5m error threshold and bottom figure is for 0.25m error threshold. Different colors shows different grid size. (a) 2.0m resolution. (b,c) 5.0m resolution. 105

Figure 7-14 laser range vs localization error. As the laser range increases the localization error decreases. 107

Figure 7-15 Matching time vs laser range. As the laser range increase the matching time also increases. 108

Figure 7-16 Average number of points in each input scan of laser scanner vs laser range in Path I. As the laser range increases the number of points also increased. Red graph is for the input scan after down-sampling with 1.0m grid size. 108

Figure 7-17 Comparison between the number of points and matching time. 109

Figure 7-18 average number of points in each input scan of laser scanner vs laser range in Path I. as the laser range increase the number of points also increased...109

Figure 7-19 Optimized laser range for different sample points in the map for two error threshold. In top chart (red), laser range is selected so that the error become less than 10cm and in bottom chart (red), error threshold is 25cm...110

Figure 7-20 Error modeled from map evaluation factors...112

Figure 7-21 Adaptive range determined by comparing to map evaluation factors. (Top) for error threshold of 10cm. (Bottom) for 25cm error threshold. 112

Figure 7-22 Adaptive range determined by map evaluation factors for 10cm error threshold. 113

Chapter 1.

Introduction

The autonomous vehicle has gained a focus as an essential technology within intelligent transportation systems since its introduction in the early 1950s. Since then, many attempts has been made to bring this intelligent vehicle on the roads fully or partially to benefit from its capabilities. In order to make this technology work in the real-world conditions as well as the controlled environment in laboratories, researchers have encountered many challenges. One of the most persistent challenge has been the vehicle self-localization.

The vehicle self-localization, also known as the ego-vehicle positioning, is a crucial technology to assist with keeping the vehicle on the road, within the correct lane, being aware of the stop lines, and providing accurate navigation options. In order to achieve that, the vehicle should be able to know its lateral and longitudinal position within decimeters. On the other hand, high precision self-localization solution when combined with prebuilt map can simplify the difficult concept of perception and scene-understanding into a less complex positioning problem. In this case, position of static obstacles such as buildings, traffic lights, zebra crossings as well as following the traffic rules can be retrieved from ego-position rather than complex scene understanding methods (Figure 1-1).

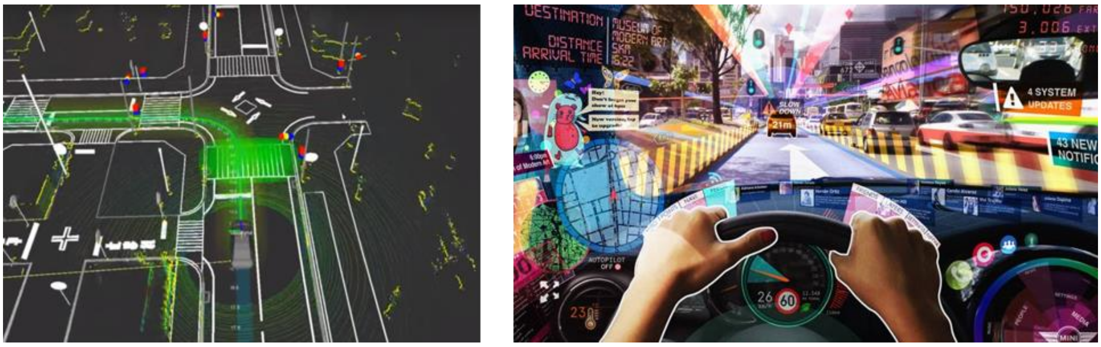


Figure 1-1 Assist of self-localization for various application such as augmented reality (AR).

However, this level of accuracy is hard to achieve by GPS-based methods in urban areas due to signal blockage and multipath effect. In recent years, use of Light Detection and Ranging (LiDAR) technologies for self-localization become more popular due to its price drop and miniaturization. LiDAR-based localization methods can be divided into two main categories. Map-based and without map. Chapter 2 of this dissertation is covering the state of the art vehicle self-localization methods using variety of sensors including GPS, LiDAR, and vision.

Generally, in LiDAR-based localization methods, a map of the environment is made beforehand using high-end mapping systems. In the localization phase, a 360˚ scan of the environment captured by LiDAR is matched to the map to get the accurate position of the vehicle. For accurate self-localization, the global and local accuracy of the map is essential, and many types of research have been done to obtain such a highly accurate map. One of the challenges of the mapping is an accurate localization of the mapping system in the urban environment. In the urban environment the NLOS and Multipath effect cause the localization with the GPS system infeasible. Even real time kinematic (RTK) one. Thus, the location of the vehicle has error and as a result the global and local accuracy of the map degrades. Local accuracy degradation makes the precision of the map very low. Low precision means that for one single object in the map, several map object with inaccurate location appears. For example for one single poles, several poles arear. This will cause error for the methods of self-localization that later uses this map. In order to get the accurate position of the vehicle in such a system high-end IMU are used. Even with the high-end IMUs the location accuracy of these systems are highly influenced in urban environments where many tall buildings surrounded the vehicle. As an alternative way, the generated map are calibrated using hundreds of ground control points (GCP). This work is very labor and time intensive. For example, for each square kilometer, around 244 GCP is used [2]. Obtaining the GCP is very labor intensive. In order to reduce the labor and time complexity, airborne imagery is used an extra data source are used [3], [4]. In [2] generated map is post-calibrated by airborne imagery automatically. The lane marking from the point cloud map and airborne imagery are both extracted and matched together. By doing this, the generated map can be calibrated to the global coordinate. Utilizing this idea, we have developed a mobile mapping system which uses the airborne image as a complementary data sources to update the trajectory of the vehicle. By updating the trajectory of the vehicle, with globally accurate airborne image, the accuracy of the map generated from this mapping framework are guaranteed. This mapping system utilizes two laser scanner. One is used for SLAM and the other is tilted downwards to capture the environment as a map. In Chapter 3, urban mapping frameworks and the results are presented. All map used in this dissertation is made by this mapping framework. Proposed mapping system and generated map using this system is shown in Figure 1-2.

One of the challenges for using point cloud map for localization is its size. The size of point cloud is very heavy. For a small area, for instance 300m to 300m, around 250 million points should be stored to the autonomous vehicle storage system. On the other hand, the matching algorithm requires high computational time to deal with this data size. In order to reduce the size of map, and yet keep the valuable features for matching, two self-localization method with two map abstraction method is proposed in Chapter 4. These two map formats are: multilayered 2D vector map, and probabilistic planar surface map. In the multilayered 2D vector map, which is used for 2D positioning, different layers of building footprints are stored as a vector in the map. The vectors are extracted from Point cloud map. By using different layers of the building, the feature for matching increases and make the matching more accurate. For each vector, the variance of the points are stored in the map to allow the localization process to rely on more certain vectors rather than an uncertain one. This can reduce the map size to several thousand

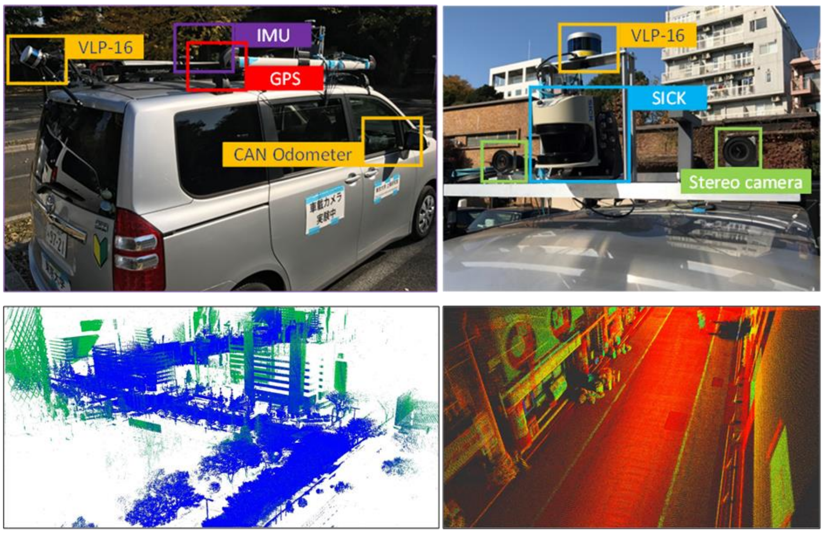


Figure 1-2 GCP extracted for 1km^2 to post calibrate the MMS scan

times. Localization is performed by maximizing the likelihood function between input scan and normal distribution made from the vectors. As the normal distribution is made from the vector, the discretization makes less uncertainty and thus the localization accuracy increase comparing to the 2D point cloud map. On the other hand, in the probabilistic planar surface map, instead of using heavy point cloud, the planar surfaces are extracted from the point cloud map. Each planar surface contains the variance of the points which shows the uncertainty of the planes. Storing the variance for each plane help the localization process to rely more on the certain walls. In the matching phase, distortion of the input scan is one of the challenges which makes the misalignment of the input scan to the map [5]. In this work, distortion is removed in the optimization process. In fact, in each iteration of the optimization, the scan is reshaped by the new results and the distortion is removed to fit better to the map. Overview of these two localization method is show in Figure 1-3.

In the map based categories, map plays a significant role in achieving high accuracy self-localization. For accurate self-localization, the global and local accuracy of the map is essential as well, and many types of research have been done to obtain such a highly accurate map [3], [6]–[8] including the mobile mapping framework proposed in Chapter 3. However, the highly accurate map does not guarantee the accuracy of the localization [1]. In other words, map accuracy is different than the ability of the map for localization. For example, in the case of the tunnel, no matter how much the map is locally and globally accurate, the lack of longitudinal features in the map causes localization error in moving

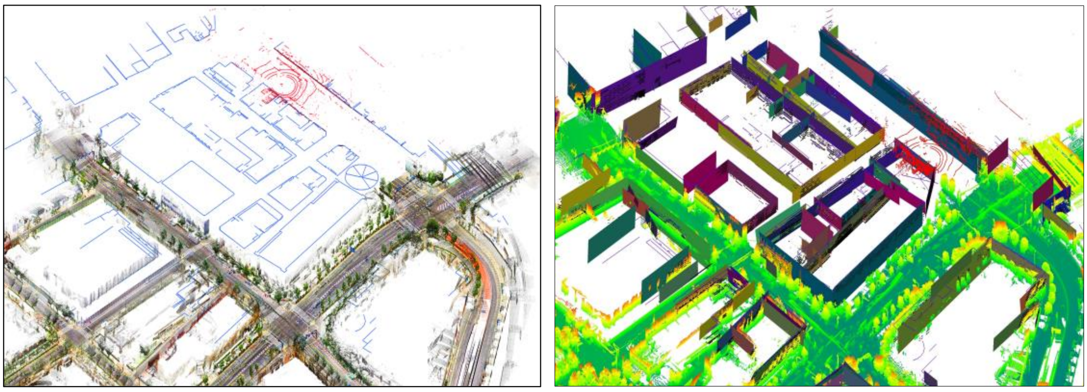


Figure 1-3 Overview of abstract map based localization techniques. Left image shows the multilayerded 2D vector map method. The right image shows the probabilistic planar surface map based localization.

direction.

To achieve accurate self-localization within a map, the map should satisfy some requirements. In other words, the map should meet some specific criteria which define the ability of the map for self-localization. To the best of this author's knowledge, there is no comprehensive study of the definition and formulation of these criteria. Therefore in Chapter 5, the required criteria regarding the ability of the map for accurate self-localization are defined and formulated.

Some of these criteria highly related to the environment. For example, in the city environment, there are more structured artifacts than rural places or crop fields. Thus the features for the map-matching can be found easier, and as a result, the localization become more accurate [9]. Tunnels, urban canyons, and highways can be assumed as another example in which environment are not suitable for self-localization using map-matching. In these scenarios, the vehicle is surrounded by two long walls in each side. Thus, the features needed for longitudinal positioning are not enough, and as a result, there will be an error in moving direction. Surrounding environments are different from place to place in the map and should be evaluated by defined criteria.

Some other criteria are related to the quality of representation of the environment by the map. In addition to the sensor related parameters in the mapping phase such as frequency of laser scanner, layer count, the range of the beam, setup parameters, etc., quality of representation highly related to the format and abstraction ratio (resolution) of the map. Many types of research have been done to propose an abstracted map format to both reduce the map size and the computational complexity [10]–[17]. Quality of representation of the environment in each of the map formats are different. In other words, some map formats discards more details of the map comparing to others. This information loss of the map might change the quality of some of the criteria and lead to localization error [18]. However, in some part of the map, abstraction does not necessarily change the quality of the map, or rate of change is acceptable. Additionally, in some cases, other criteria might compensate the lack in one criterion. Therefore, in order to evaluate the self-localization ability of the map in a specific point, all criteria should be considered together.

In Chapter 5, three general criteria for map factors are defined. These criteria are

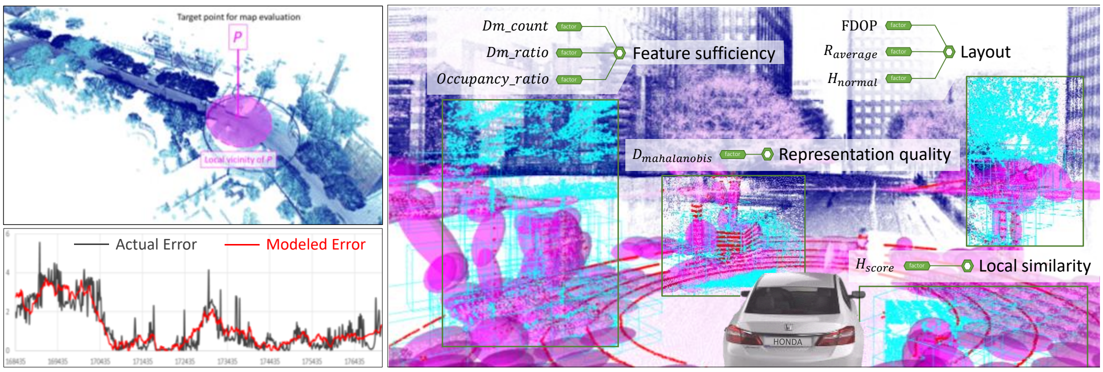


Figure 1-4 . Overview of the proposed map evaluation method.

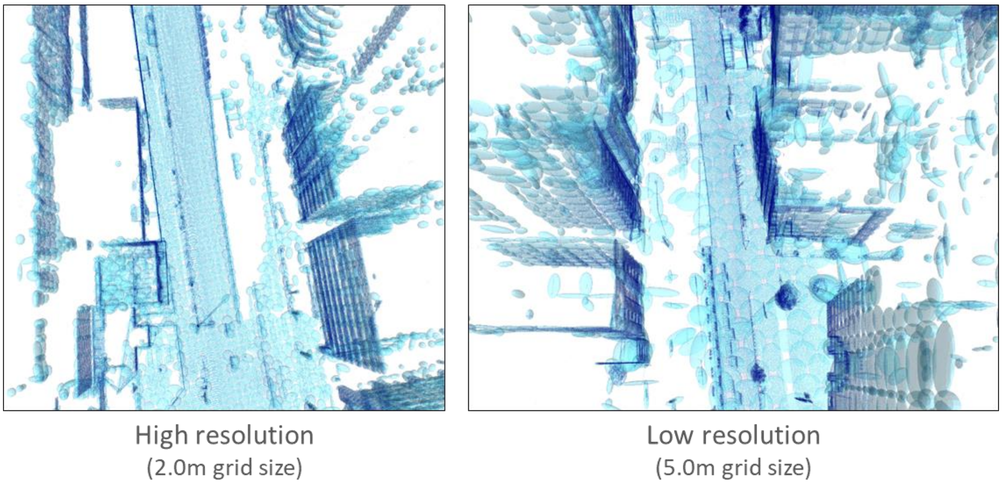
feature sufficiency, layout, and representation quality of the map. These criteria are defined by using top-down and bottom-up approaches iteratively. Each of these criterion answer one of the question about the map such as "*are the features enough for localization?*", "*are the buildings' layout suitable?*", "*does the representation quality enough?*" and etc. These criteria are defined regardless of the map format and can be applied to any other map formats. For quantifying each of these criteria, in chapter 5, the focus is made on the normal distribution map format, and several factors are defined. For each point in the map, these factors are calculated based on the features in surroundings called local vicinity. By obtaining the correlation of the map factors with localization error, the effectiveness of the factors is investigated. By comparing the results, one more criterion which is local similarity is added to the criteria list and for quantifying the local similarity, three factor are introduced. These factors are score entropy, point feature histogram (PFH) based similarity and Battachariya distance of the features. Chapter 5 only introduces the criteria, and factors.

In chapter 6, the effectiveness of each factors are investigated and experimental results are shown. The experiments have been conducted in the Shinjuku, Tokyo, Japan, which contain around 40km of route. Additionally, by applying principal component regression (PCR), the predictability of the self-localization error based on these factors are investigated. In Chapter 6, effectiveness of each factors are investigated by accumulatively adding the map criteria to the model and comparing the goodness of fit results. Figure 1-4 shows the overview of proposed map evaluation frameworks. In addition to PCR, in order to increase the capability of modeling the error based on the defined factors, and also increase the possibility of combining these factors together, simple feedforward neural network is used. The input to the network are defined map factors and the output is error rescaled to 0 to 1.0.

In chapter 7, possible outcomes of the map evaluation factors are proposed. One of the highly valuable outcome of this study is the ability of adaptively determining the resolution of the map. In chapter 7, effect of map resolution on the localization is deeply discussed. Basically lower map resolution (more abstraction) is more desirable as it can save the computation time and memory size, but it might remove some important information for localization. In order to determine the best resolution of the map, the map is evaluated for different resolution using map evaluation factors and the optimum resolution is selected. By doing this, the map resolution can be determined adaptively based on the presence of the features for map-matching (Figure 1-5).

On the other hand, the range of the laser scanner is an important parameter for reducing the matching time. This parameter can be determined adaptively based on the quality and quantity of the features in the surroundings and stored in the map beforehand. In Chapter 7, adaptive determination of the laser range based on map factors are proposed which is another outcome of this dissertation. In this method, the optimum laser range are acquired from the map factors and stored in the map. And finally, chapter 8 concludes this dissertation and discuss the possible future works.

In the Appendix I, the project of graduate program for social ICT Global Creative leaders (GCL) are explained.



High resolution
(2.0m grid size)

Low resolution
(5.0m grid size)

Figure 1-5 Different map resolution obtained for different part of the map by adaptive refinement method

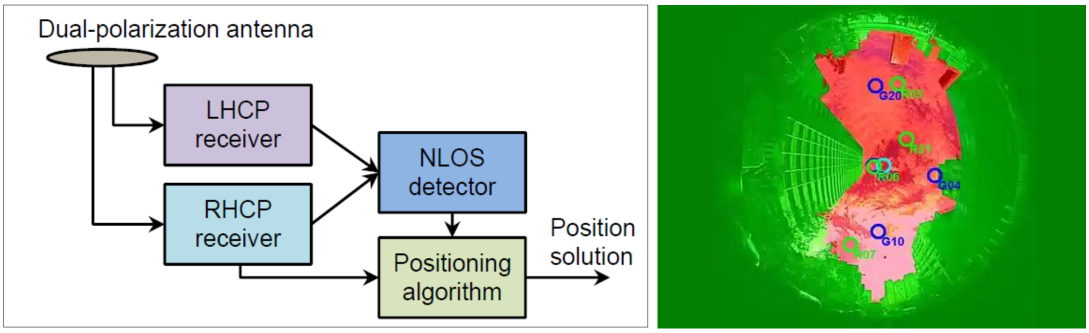
Chapter 2.
Vehicle self-localization and digital map

For acquiring precise ego-vehicle position, variety of technologies such as Global Navigation Satellite System (GNSS), Inertial Measurement Unit (IMU), monocular and stereo vision, single and multi-layer laser scanner have been studied so far. Among them, GNSS is the most widely used technology which provides an affordable way to accomplish worldwide positioning. For the intelligent vehicle self-localization, there are several metrics such as accuracy, robustness, availability, computational cost, and memory cost.

Papers [19]–[29] investigate GNSS based self-localization. Since the reliability and accuracy of GNSS alone is not sufficient for autonomous driving even in an open-sky environment, papers [19]–[21] took the advantage of adding IMU to GNSS to obtain better accuracy. In [19], by modeling the residual correlated error with Parallel Cascade Identification (PCI), more accurate pseudo ranges are obtained and tightly coupled with 3D Reduced Inertial Sensor System (RISS) with mixture particle filter (M-PF). In [20], besides the IMU, velocity and yaw angle are fused with Global Positioning System (GPS) using a novel parallel-dual-H-Infinity filtering (PDHF) mechanism. In [21], momentary blockage of satellite signals caused by utility and light poles in relatively open sky is distinguished by analysis of rapid changes in navigation satellites' signal strength.

In contrast to the open sky scenarios in the urban environments, because of various obstacles and tall buildings, satellite's signals are blocked or reflected. This reflection and blockage of signals which cause non-line-of-sight (NLOS) and multipath reception, dramatically degrades the accuracy of positioning. Paper [22]–[25] investigate a solution to this problem in urban environments. Papers [22], [23] improve the positioning accuracy by removing the erroneous signals. In [23], high-mask-angle antennas (HMAAs) are used to mitigate interference signals coming from a low-elevation angle and the optimal mask angle for multi-constellation is obtained. In papers [21], [24], [25] geographical information such as 3D building model [25], urban trench model [24], 3D shape of road signs and bridges [21] are used to mitigate the effect of multipath and NLOS signals. In [26], 3D map of the drivable area is tightly coupled with GPS pseudo-ranges to obtain better accuracy. In paper [27], precise vehicle side slip is obtained by INS/GPS integration. In [30], [31] the pitch model and road slope are used for localization. In [30], the effect of road slope on position estimation is studied and geometric information of road, specially road slope, was utilized for vehicle position estimation to acquire better positon accuracy. In [31], the road data is encoded using linear dynamic models and the position of the car while traveling is obtained by the continuous comparison of a bank of linear model. In Figure 2-1(a) dual polarization antenna is used to detect the NLOS. Usually signals from satellites are right handed. If the signal is reflected by building the polarization is changed. So by comparing the polarization of the incoming signal we can detect if it is reflected signal or not. Figure 2-1(b) shows the sky-pointing camera observation. By using the sky-pointing camera, the building are detected and the position of the satellites are projected to the camera image. By comparing the building position and satellite position we can detected if the satellite is occluded or not. If the satellite is occluded and the signal from that satellite is received, it means that the received signal is NLOS signal.

Papers [32]–[41] investigate the positioning solution utilizing vision/camera technology. Paper [32] describes the role of machine vision in intelligent vehicle which covers role of vision and maps in self-localization as well. In papers [33]–[35] prebuilt map and vision is used for localization. In [33], image from onboard monocular camera is matched with visual map which is generated by stereo vision beforehand. In [35]



(a) Dual polarization antenna

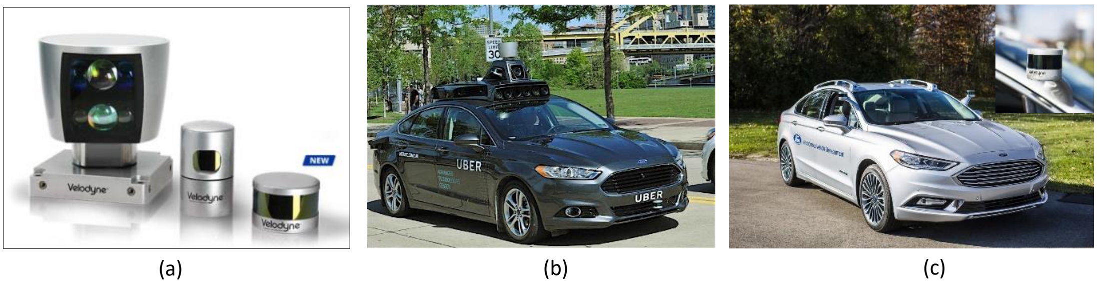
(b) Sky pointing camera

Figure 2-1 GPS NLOS detection. (a) Using dual polarization technique the NLOS signal is detected [31]. (b) Using Sky-pointing camera the occluded satellite are detected [33].

localization in intersection using stereo vision and detailed digital map consisted of lane markings and curbs are studied while in [34], in addition to front camera, rear and side cameras' images are also matched with the digital map to localize the vehicle. In order to take advantage of equipment already deployed in the car such as infrared camera (IR), [36] extends the concept of visual odometry (VO) to multispectral odometry (MO) by computing ego-vehicle position estimation from cameras working in different spectral bands. In [37], laser and camera were combined to calculate the relative displacement of the vehicle instead of IMU which suffers from drift. In papers [38]–[41], lane detection algorithm with vision sensor is utilized to obtain a lane level position of the vehicle. In [38], lane detection algorithm is assisted by the vehicle detection and vice versa.

Aforementioned localization methods are only based on the data from their own sensors. On the other hand, in [42]–[50], inter vehicle communication technology is used to share the position-related information between vehicles and perform localization task cooperatively. In paper [42], each vehicle shares their IMU and INS-based position data with each other and collective data are used to obtain accurate position in difficult GNSS environments including deep urban canyons and tunnels. In [43], [44] GNSS pseudo-range corrections are generated cooperatively between the vehicles instead of fixed stations. In [44], by having a precise map and exchanging the GPS raw measurements, cooperative map matching is performed. Papers [45], [46], [48] investigate relative positioning in a vehicular ad-hoc network (VANET). Finally in [51], [52] GSM and cellular networks are used to assist the self-localization.

LiDAR is another technology that have gained the focus to be a key technology for



(a) (b) (c)

Figure 2-2 Velodyne laser scanner for different autonomous vehicle platform. (a) from left, a 64 layer Velodyne laser scanner (HDL-64), 32 layer (32-DL) and 16 layer (VLP-16). (b) HDL-64 is mounted on Uber autonomous vehicle. (c) VLP-16 is mounted on Fords autonomous vehicle.

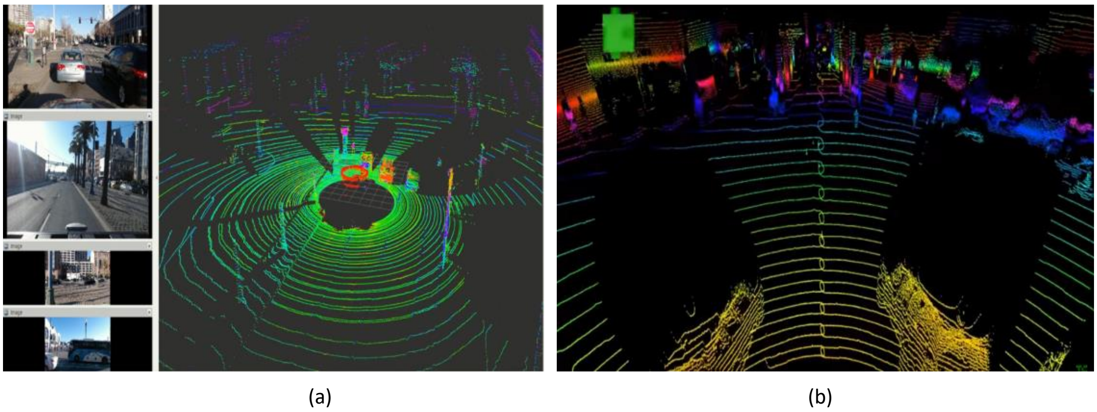


Figure 2-3 (a) Velodyne HDL-64 scan data which is captured in California, U.S. (b) Data from Luminar laser scanner. Luminar laser scanner does not rotate and it is stationary.

perception in autonomous vehicles. Current laser scanners are too bulky and expensive to be deployed in the intelligent vehicles. However, many companies have already started the miniaturizing and lowering the cost of this sensor to be able to be mounted on the intelligent vehicles. In Figure 2-2 various type of laser scanner are shown. Laser scanner has different field of view. Some of them can rotate 360 degree and collect the data from all 360 degree. Some of them only capture part of the horizontal view. Number of layers in the laser scanners are different. Some of the laser scanners only have 1 layer. Other types has multiple layers. For example Velodyne laser scanner have multiple layer. It has 16 layers model, 32 layers model, 64 layers model and recently it introduced a 128 layers model. The frequency of laser rotation or capture also varies from model to model. Some models capture data with 5Hz. Some model such as SICK LMS-511 PRO can collect the data with frequency of 100Hz. The frequency is important for the data density and distortion while vehicle moves. If the frequency of capturing data is low, then the distortion in the laser data will be high. If the frequency of the laser is high, then the distortion in the laser data while vehicle moves are high. The data collected by Velodyne 64 layer laser scanner (HDL-64) is shown in Figure 2-3.

In [23] the author, for the first step apply scan-to-scan matching with ICP to get initial guess for the next step. Next step is matching scan to map using NDT. NDT has problem with initial pose so with this technique we can overcome this problem. In this NDT each cell is divided to multiple normal distribution to match the points better. Also the new scan is represented as ND. The author said that this (ND-to-ND) technique is novel but before that Magnusson proposed this and also open source are available in ROS. For matching the similarity of normal distribution Kullback-Leibler is divergence is used.

- Feature-based

Feature-based SLAM techniques, a set of features including different types of distinct geometric models such as points, lines, curvatures, and any arbitrary shapes are extracted from the observation and used as landmarks to associate the new and previous observations. Finding landmark is not so easy.

- Scan-based

Directly used unprocessed scans from laser scanner. Association or matching of current scan to previous scan Scan-to-scan matching.

There are several methods for scan-to-scan matching.

- ICP iterative closest point (the most common) [34]

Point to point matching between two scan by minimizing the total distance between them. May lead to inappropriate data association because cannot find corresponding point sets

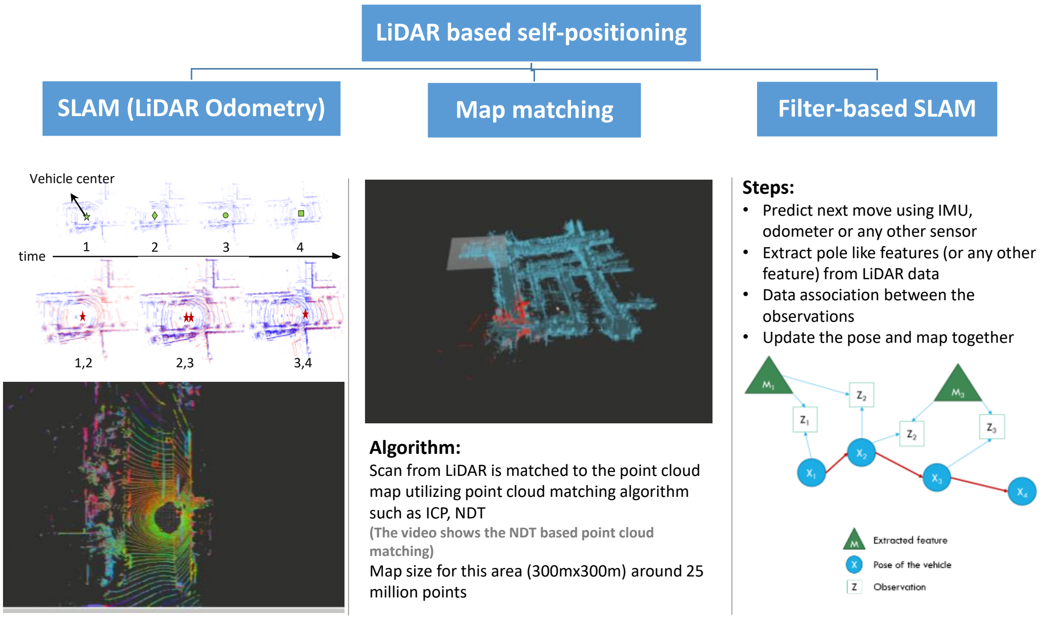
- In [45] angle between two points are used for matching

Weiss and Puttkamer [37] proposed a technique that avoids the point-to-point matching problem by calculating an angle between two neighboring scan points for every scan point and using the angles to match the two scans.

- NDT
- Lu and Milios [43], which performed the matching of the new scan to the previous scan and further matched all the scans by storing the past scans
- Thrun et al. [23] used the expectation maximization (EM) algorithm that finds the best matching past
- Due to the need for matching to all the past scans for the best accuracy, Bosse et al. [16-in the docs] introduced a subspace-to-map matching technique where the new scan is matched to all the past scans of a subspace of concern with any scan-to-scan

matching technique and the subspaces are subsequently associated to each other for global mapping. This technique achieves the matching of the new scan to all the past scans, but the accuracy could still drop since the new scan points not in the subspace are not matched to the past scans.

In paper [17], [53]–[56] LiDAR based localization techniques are studied. Generally LiDAR based positioning techniques can be divided into map based and without map. In no-map categories, prior map is not available. The vehicle moves and make map real-time and at the same time use that map for positioning. This technique called Simultaneous Localization and Mapping (SLAM). SLAM techniques can be divided to scan matching and filter based methods. In scan matching methods which is also known as LiDAR-odometry, the consequent scans from laser scanner are matched to each other and the movement of the vehicle is calculated. In the filter based SLAM, first, from the laser scanner data specific features such as poles are extracted. Next, position of these objects are stored. After vehicle moves, again the same features are extracted from new scan. By comparing the positon of the features in both scans, the displacement of the vehicle are calculated. Papers [53], [55] investigates the simultaneous localization and mapping (SLAM) using LiDAR. Paper [54], [56] extract the curbs or road markings from



LiDAR based self-positioning

SLAM (LiDAR Odometry)

Map matching

Filter-based SLAM

Algorithm:
Scan from LiDAR is matched to the point cloud map utilizing point cloud matching algorithm such as ICP, NDT
(The video shows the NDT based point cloud matching)
Map size for this area (300mx300m) around 25 million points

Steps:

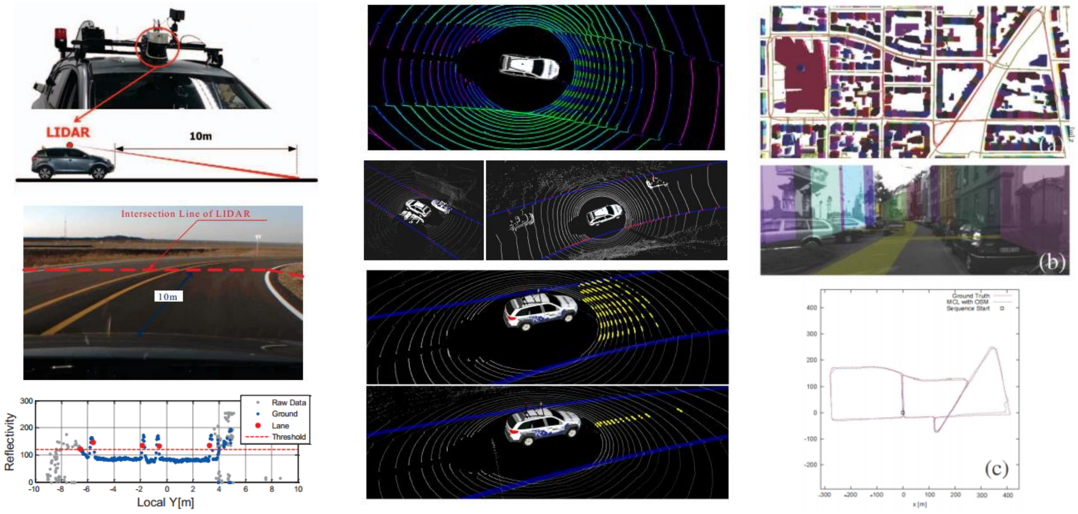
- Predict next move using IMU, odometer or any other sensor
- Extract pole like features (or any other feature) from LiDAR data
- Data association between the observations
- Update the pose and map together

Figure 2-4 Lidar based self-localization categories. SLAM, Map matching.

LiDAR point cloud and use it for positioning. These categories are shown in the Figure 2-4.

SLAM based methods suffer from accumulative error. In each step of positioning there is a small error. As the position is not updated by any reference this error increase the total error. Therefore map based methods are more reliable.

In the map matching based localization methods, map is generated beforehand and later using the map and matching of the laser data to that map, the position of the vehicle are obtained. There are many types of map such as point cloud map, lane markings map, corner map. For example, in [29] the road marking map is generated from laser scanner data beforehand and later by matching the laser data to the road marking map, the position of the map is generated. Using laser scanner, and intensity value of the laser beams, extraction of the road marking become more accurate and reliable. This method is shown in Figure 2-5(b). In [33] the localization is performed by single layer laser scanner by detecting the curbs and lane markings and matching it to the roadmarking map. This method is shown in Figure 2-5(a). Paper [39] investigates the impact of different LiDAR observations from the upper and lower layers of a multi-layer laser scanner in highway and urban scenarios. One of the map type for self-localization is 3D point cloud. Sample of the 3D point cloud map is shown in Figure 2-6.



(a) (b) (c)

Figure 2-5 Methods which is used laser scanner and prior map for self-localization [29] [33].

In [57] camera is use to perform the localization within in the LiDAR map. This paper proposed a methods which utilizes 2D vector map (simplified 2D line feature map representation) and 3D laser scanner and GPS and IMU and odometer

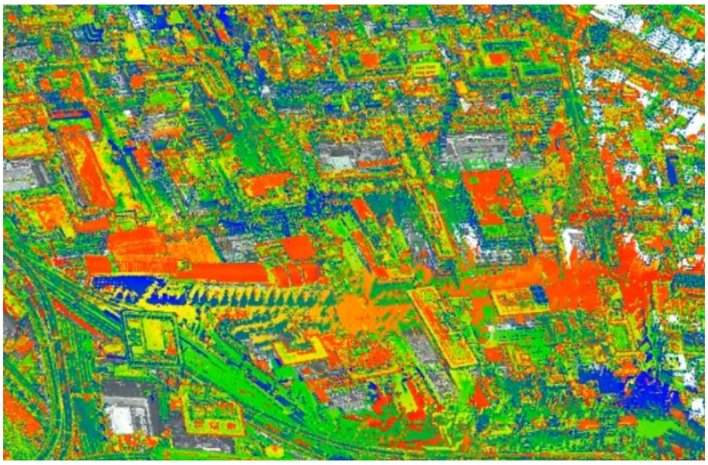


Figure 2-6 Point cloud map of the Hitotsubashi, Tokyo, Japan. This point cloud is the combination of the (Mobile Mapping System) MMS data and airborne laser scanner (ALS) data.

Chapter 3.
Low-cost urban mapping framework

3.1. Introduction

As described in Chapter 2, the most reliable and accurate localization techniques in urban environment is based on point cloud map matching. In this method point cloud map should be generated beforehand. Usually this point cloud map is generated by the Mobile mapping systems (MMSs). MMSs are a sensor platform mounted on the roof of a ground vehicle in combination with high-end GPS, inertial measurement unit (IMU) and odometer to rapidly capture a lot of 3D datasets essential for creating highly accurate and high-resolution representation of the roads and their surrounding environment. These

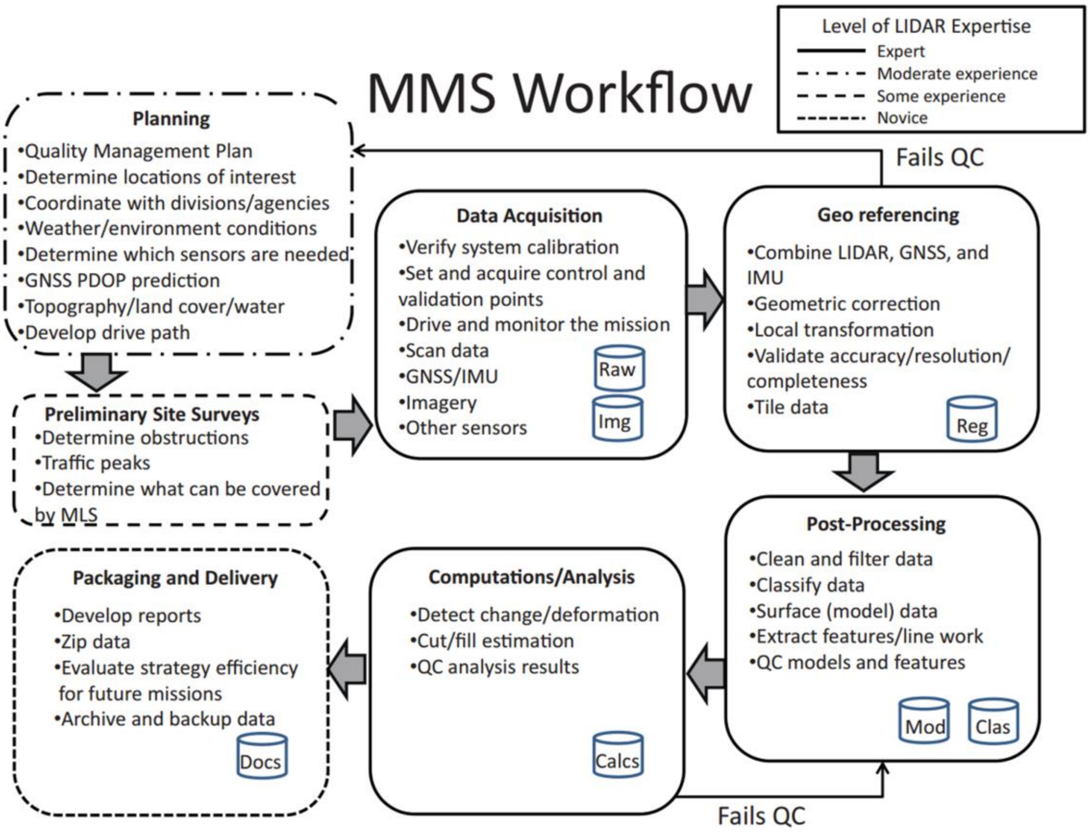


Figure 3-1. Generalized MMS workflow [13].

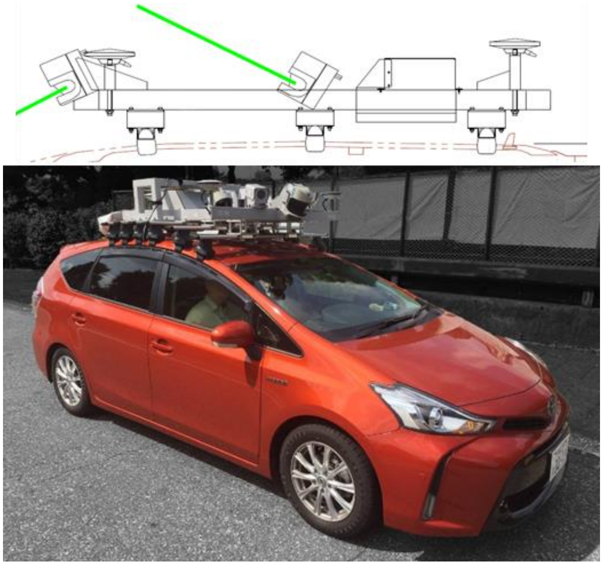


Figure 3-2. MMS which is used in part of mapping of the experimental area

systems were designed for the first time in the late 1980s when the GPS technology become available for public civil usages [1]. While the definition of the MMS is quite simple, the real MMS workflow contains plenty of essential factors which missing each of them leads fail in the quality control. Figure 3-1 illustrates the generalized workflow of the MMS including middle products of each step [2]. This workflow shows the detailed consideration of a single data collection using MMS. Traditionally, the MMS workflow consists of plenty of site surveys and experiments which require labor-intensive and time-consuming procedures. The Figure 3-2 shows MMS which is to generate map for some part of the experiments.

3.2. Mapping Platform

As the price of these system are very expensive (several hundred million yen), the low-cost mapping frame work are proposed in this chapter. In fact, the problem of price of MMS is coming from two factors. One is the price of sensors and technologies, and the other is the labor fee for post-calibration of the map. Low cost system to make the point cloud map by integration and use several data sources. The sensor used for mapping is as follows:

- SICK laser scanner (25 Hz)
- Horizontal Velodyne laser scanner (10 Hz)
- Tilted Velodyne laser scanner (10 Hz)

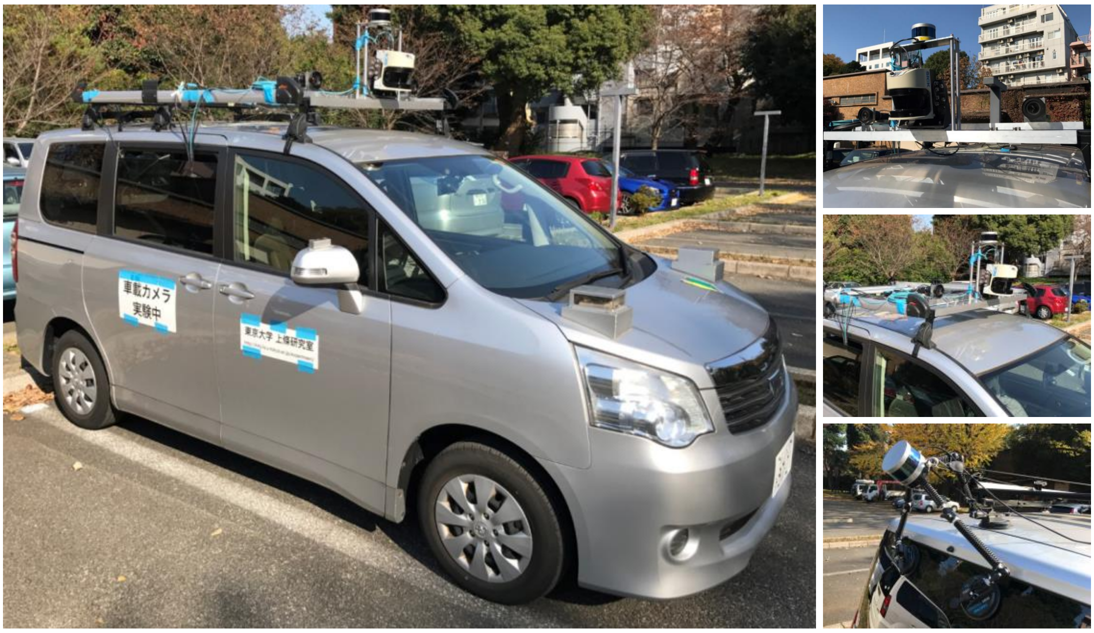


Figure 3-3. Proposed mapping system

- IMU (8.4 Hz)
- Odometer (8.4 Hz)
- Differential GPS (1 Hz)
- Stereo Camera (10 fps)

The experimental vehicle for the map mapping and the employed equipment are shown in Figure 3-3 and Figure 3-4. The single layer SICK laser scanner is installed in front of the vehicle and tilted to see the ground. Two side of the SICK is the stereo cameras which can be used for localization, mapping and perception. There is one horizontal Velodyne and one tilted one, which the first one is used for the localization and fused with

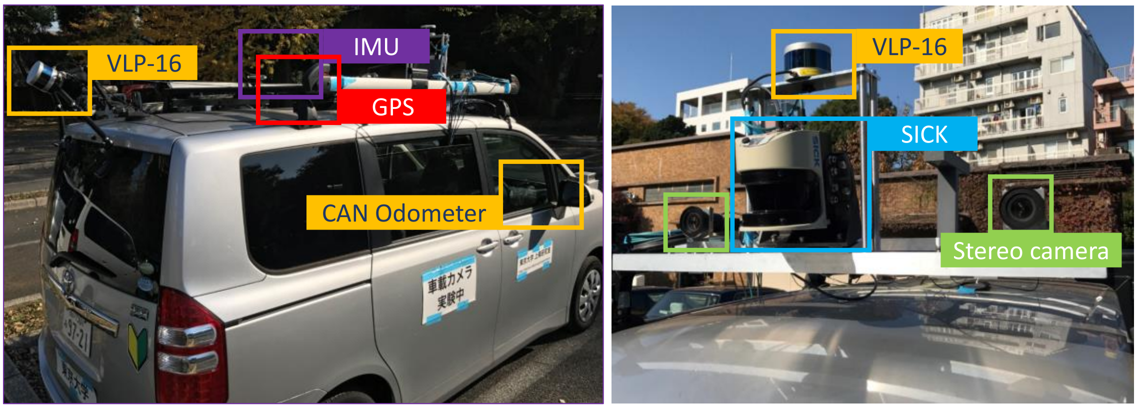


Figure 3-4. Equipment which is used for mapping

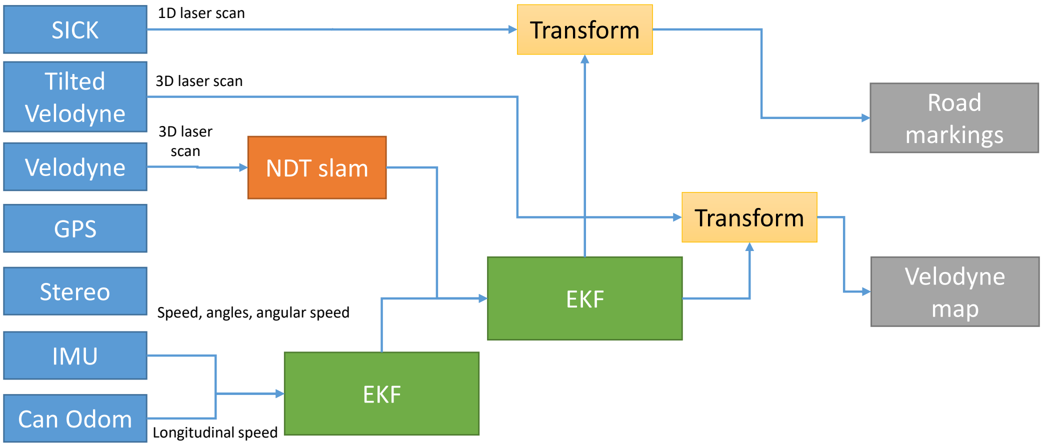


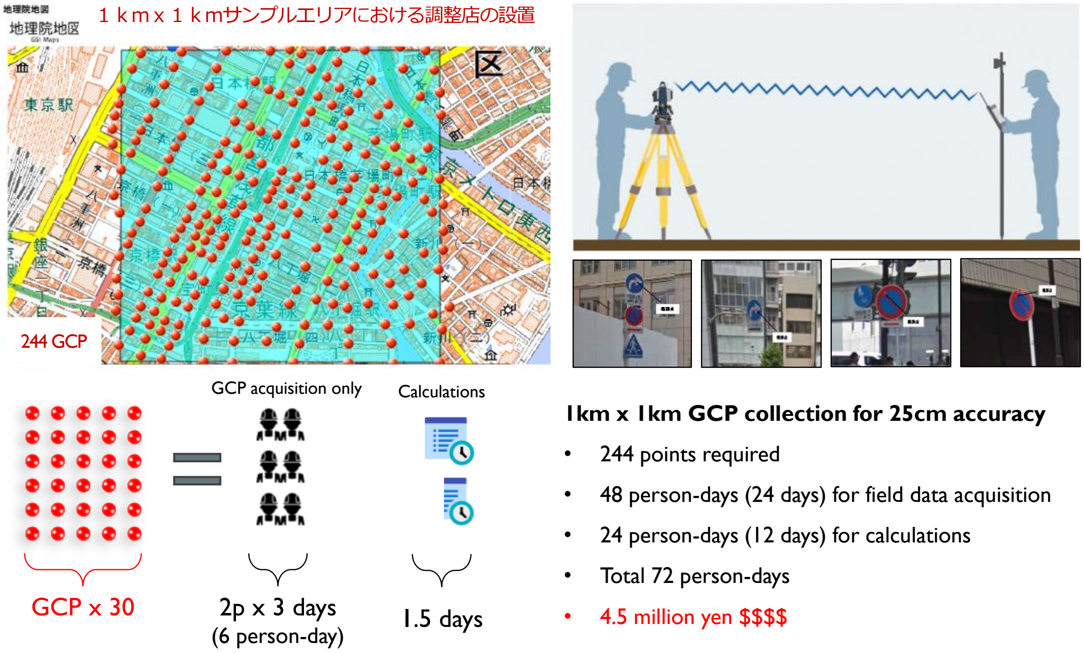
Figure 3-5. Equipment which is used for mapping

other sensors and the other one is used for the mapping. IMU sensor on center of the roof and the odometer read from the CAN system is also an essential component to be fused for the vehicle localization. The mounting system is customized by the factory to be solid enough for the experiments since it is important in the fusion to know that the relative location of the sensors are fixed and not changing during the experiment.

3.3. Mapping Framework

All these sensor information are fused together in a double extended Kalman filter (EKF) framework to calcluate the position of the car and performed the mapping (Figure 3-5).

Even though we fuse all of these sensors, still we have a significant localization error in deep urban canyons. Therefore, the map generated using only the local sensors installed on the experiment vehicle is not enough for the autonomous vehicle's self-localization. In order to cover this problem, generally companies are collect ground control points (GCP) in the survey area (Figure 3-7) and use them to calibrate the point cloud. However, it is very labor-intensive and time-consuming procedure. The procedure of calibrating the 3D point cloud data obtained from a ground surveillance data is shown in Figure 3-8. This procedure consists of 1) handpicking the GCPs from the original 3D point cloud, 2) calculating the position correction vector (PCV) which estimates the localization error, 3) correct the ground surveillance vehicle's position using the PCV, 4) correct the vehicle trajectory using the new vehicle positions, and 5) rectify the 3D point cloud based on new vehicle trajectory. As can be seen in Figure 3-7 and Figure 3-8, the



1km x 1km GCP collection for 25cm accuracy

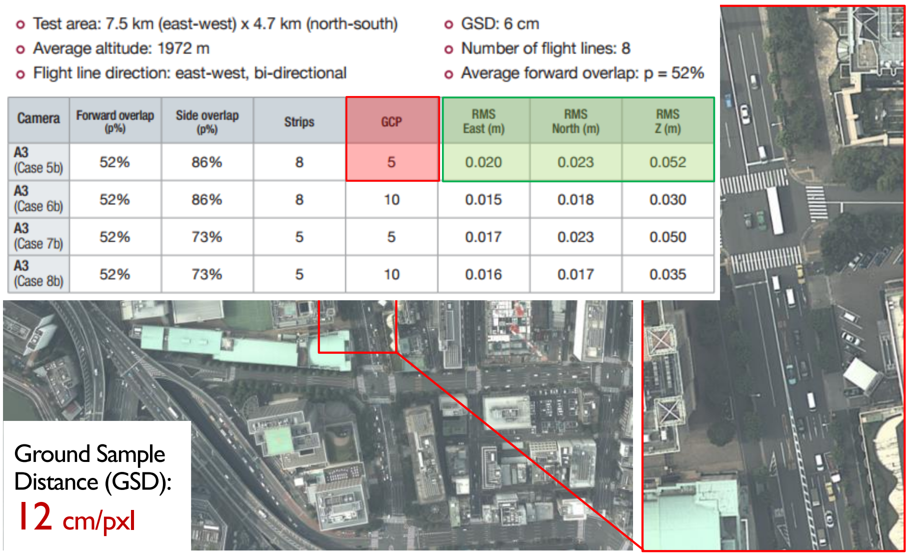
- 244 points required
- 48 person-days (24 days) for field data acquisition
- 24 person-days (12 days) for calculations
- Total 72 person-days
- 4.5 million yen $$$$

Figure 3-7. GCP acquisition in urban areas, a labor-intensive and time-consuming task

total cost of landmark updating for a square kilometer is almost 9 million yen.

3.4. Updating vehicle trajectory utilizing Aerial Data

To solve this problem and generate high-precision 3D point cloud, we extended the works [3], [4] and take into account another global information, aerial surveillance data to rectify the point cloud data acquired by our ground surveillance system. Nowadays, by

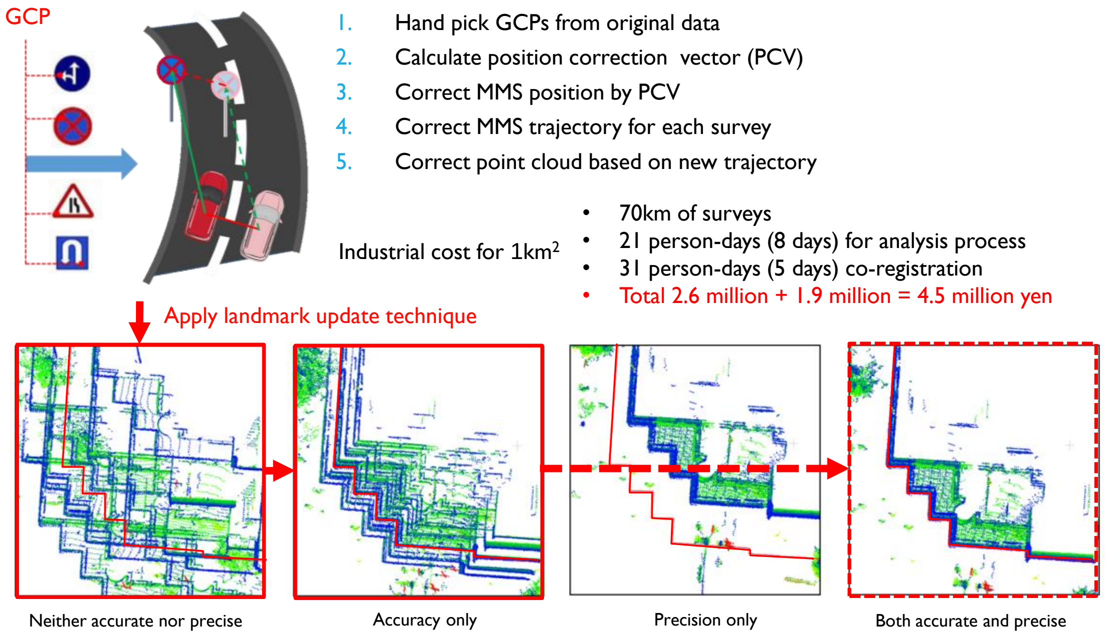


- Test area: 7.5 km (east-west) x 4.7 km (north-south)
- Average altitude: 1972 m
- Flight line direction: east-west, bi-directional
- GSD: 6 cm
- Number of flight lines: 8
- Average forward overlap: p = 52%

Camera	Forward overlap (p%)	Side overlap (p%)	Strips	GCP	RMS East (m)	RMS North (m)	RMS Z (m)
A3 (Case 5b)	52%	86%	8	5	0.020	0.023	0.052
A3 (Case 6b)	52%	86%	8	10	0.015	0.018	0.030
A3 (Case 7b)	52%	73%	5	5	0.017	0.023	0.050
A3 (Case 8b)	52%	73%	5	10	0.016	0.017	0.035

“VisionMap A3 Digital Mapping System Tested by IFP,” Test report. Available online: http://www.visionmap.com/prdPics/files/152_cs_file.pdf

Figure 3-6. Calibration of aerial image using few GCPs.


GCP
Apply landmark update technique
Neither accurate nor precise
Accuracy only
Precision only
Both accurate and precise

1. Hand pick GCPs from original data
2. Calculate position correction vector (PCV)
3. Correct MMS position by PCV
4. Correct MMS trajectory for each survey
5. Correct point cloud based on new trajectory

Industrial cost for 1km^2

- 70km of surveys
- 21 person-days (8 days) for analysis process
- 31 person-days (5 days) co-registration
- Total 2.6 million + 1.9 million = 4.5 million yen

Figure 3-8. Landmark updating technique for calibrating 3D point cloud data obtained from ground surveillance system in the urban areas.

the development in the remote sensing technologies, aerial surveillance data of the urban areas are becoming widely available with a reasonable price.

One of the characteristics of the roads and streets in the urban area is clear road markings on the road pavement. Since urban canyons are happening in the developed areas of the city, we can consider that there will be a plenty of road marking information on the pavement. As both ground experimental vehicle and the aerial surveillance data captures these markings clearly, they are a suitable information for the calibration of the map. Aerial surveillance data is capturing a large area in short time and therefore the consistency of the error is higher than ground data. Therefore, calibrating the aerial data is much easier than ground data, which is shown in Figure 3-6.

To use the aerial image to calibrate the 3D point cloud, firstly the road marking are extracted from SICK laser scanner’s point cloud. The main reason the SICK is used is to obtain high resolution road markings on the road pavement to perform this step of calibration. Then, similar road markings are extracted from aerial image. Then, the registration is performed between these two data sources by considering the aerial data as the reference. The fusion framework of the aerial image is shown in Figure 3-9. A new EKF is added to the previous filter structure without aerial calibration. In addition, we consider the distortion removal phase to increase the quality of resulting map since one of the main source of the uncertain 3D point cloud acquired from the Velodyne laser scanner is the distortion caused by vehicle movement. Using this method, the quality and accuracy of the ground-based 3D point cloud can increase to decimeter level which is accurate enough for the vehicle localization. Figure 3-11 include more details of the cheap MMS calibration method utilizing the road markings from the aerial image.

The proposed framework takes multiple MMS surveys, ALS point cloud, and overlapping high-resolution aerial images as input and provides accurate and precise MMS data as output. The proposed framework is performed in the following sequence. First, road markings are extracted from both aerial images and MMS point clouds. Next, a Gaussian mixture map is generated from the aerial road markings as a reference. Then, each MMS survey is subdivided into fixed-length short patches, and finally, the MMS surveys are aligned to the aerial reference based on the registration results from the road markings. To overcome the problem of roads obscured by buildings in the aerial images, which is called relief displacement and complicates road marking extraction, a perspective occlusion map is generated for each image using ALS and the image's position and orientation. In addition, overlapping aerial photographs (forward overlap) are considered to exclude moving vehicles which can be mistakenly extracted as road

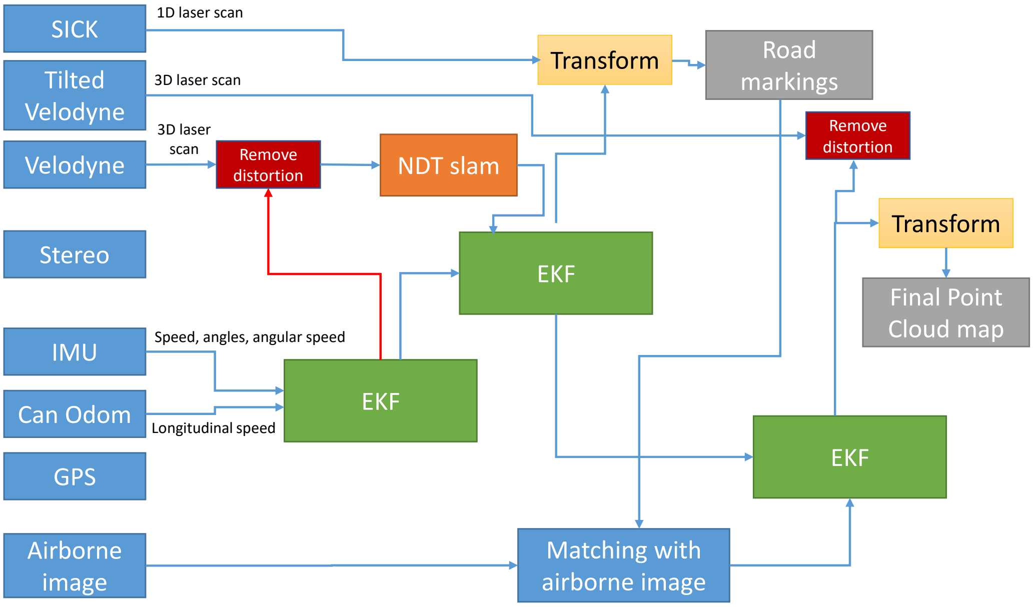


Figure 3-9. Framework of the filter including the aerial image data for the calibration.

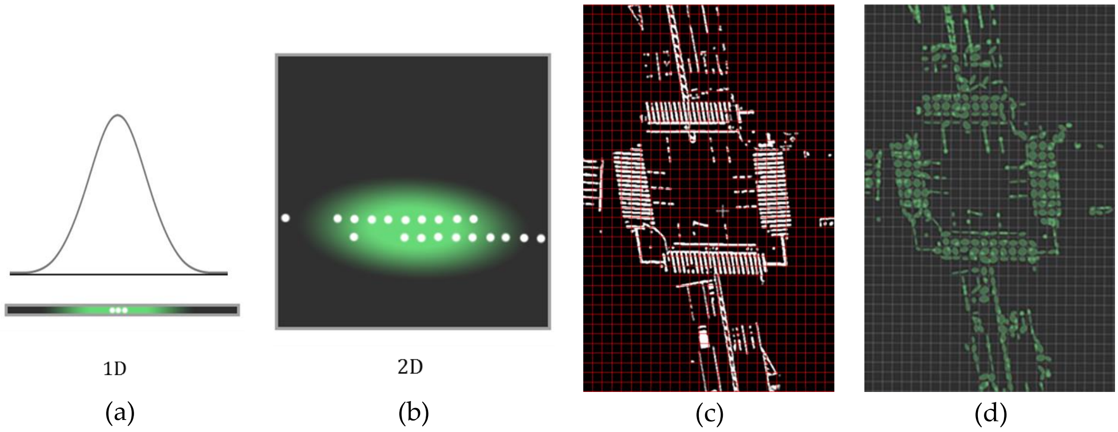


Figure 3-10. Generation of the normal distribution from the reference data: (a) Estimated 1D normal distribution of the sample points; (b) Estimated 2D normal distribution of the sample points; (c) Extracted aerial road markings; (b) Generated NDT map with 2m grid size.

features.

The road markings extracted from the aerial image are considered as references for the georeferencing of the MMS data. To obtain more features for the registration, all kinds of road markings and even signs in the sidewalks are considered based on the intensity contrast between the markings and the road surface background. After a level adjustment of the image, the adaptive Gaussian thresholding is performed to for the road marking extraction.

After the calibration of the intensity, the similar road markings from the MMS point cloud are extracted for the registration step. MMS road markings are derived by applying the adaptive thresholding. Thus far, we have obtained the road markings from both aerial images and MMS surveys. The next step is to perform registration of each MMS survey to the aerial image. We have two different input formats. To perform the registration, a dynamic-length sliding window [5] and NDT. The main idea behind the NDT is to represent the observed spatial information as a Gaussian mixture model [6]–[8]. To prepare the reference data for the registration, the bitmap image of the airborne road markings is divided into a uniform grid with a predefined cell size. Assuming that the set of all n points within the cell c_{ij}, $\{p_1, p_2, \dots p_n \mid p_k = (x_k, y_k)\}$, has been drawn from a normal distribution $\mathrm{N}(\mu, \Sigma)$, the maximum-likelihood estimate of the mean (μ) is defined as follows:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} p_i, \tag{5}$$

and the maximum-likelihood estimator of the covariance matrix (Σ) is defined as

$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(p_i - \mu)(p_i - \mu)^T, \tag{6}$$

Figure 3-10a and Figure 3-10b illustrate the representation of the estimated Gaussian distributions for 1D and 2D cells. Figure 3-10c shows a part of the aerial road markings, and the generated Gaussian mixture map is shown in Figure 3-10d.

The probability of observing a road marking at a certain 2D coordinate of the cell c_{ij} is derived from the following equation:

$$P(p) \sim e^{-\frac{\left(p-\mu_{ij}\right)^T \Sigma_{ij}^{-1}\left(p-\mu_{ij}\right)}{2}}, \tag{7}$$

where p presents the 2D coordinate (x, y) of the point.

The size of NDT grid is an important parameter for the registration. If the grid size is set to be larger than the suitable dimension, the defined probability density function for the cell might not be able to represent well the points included in the cell. In contrast, if the grid size is small, each grid can capture only a few number of input road marking points, so there will be only a few grids having the normal distribution [9]. Therefore, a small error will make the input points out of the corresponding reference grid.

By using this method, accurate and precise map can be made. Figure 3-12 shows the generated map using proposed mapping framework.

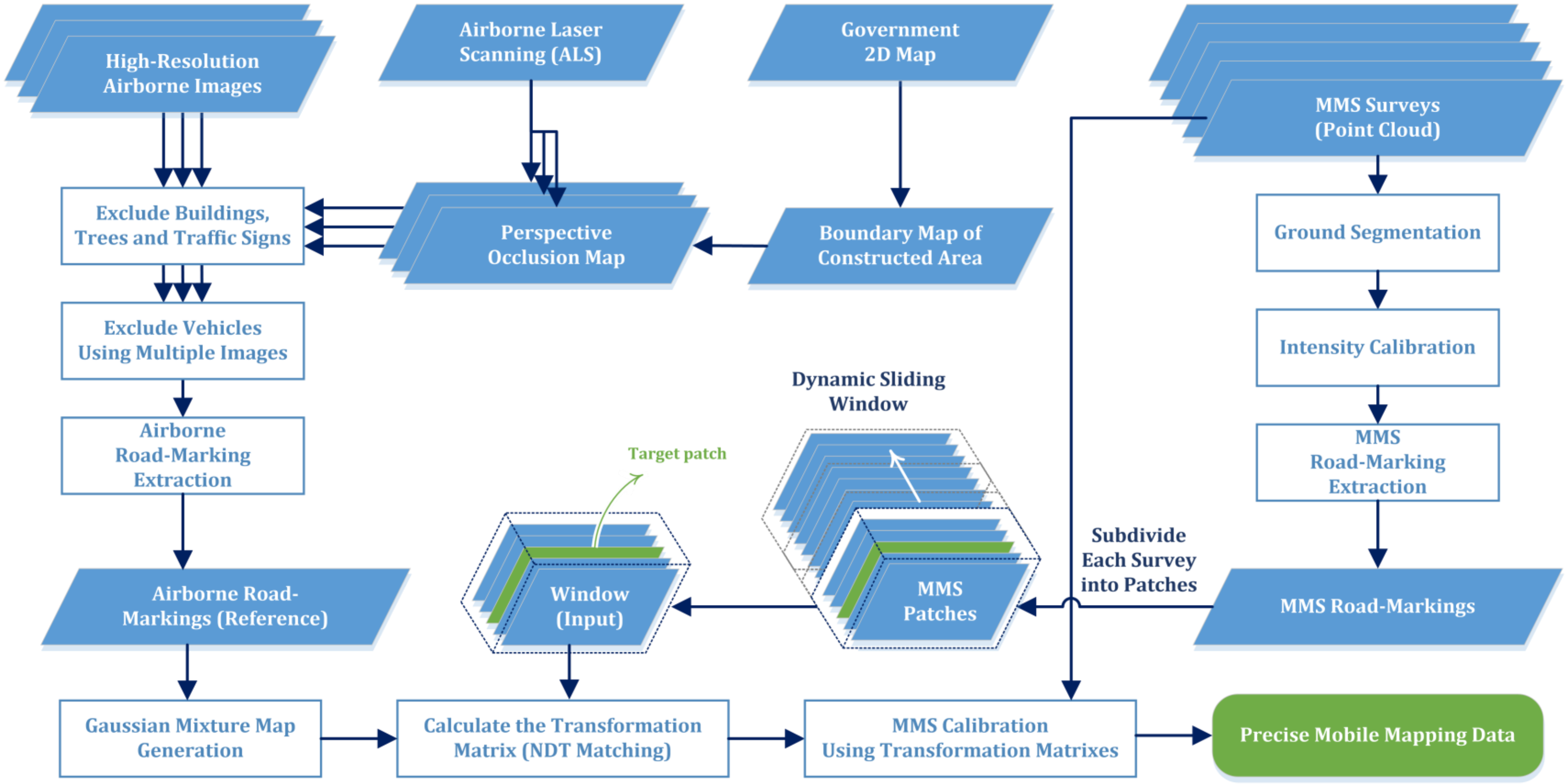

High-Resolution Airborne Images
Airborne Laser Scanning (ALS)
Government 2D Map
MMS Surveys (Point Cloud)
Exclude Buildings, Trees and Traffic Signs
Perspective Occlusion Map
Boundary Map of Constructed Area
Ground Segmentation
Exclude Vehicles Using Multiple Images
Intensity Calibration
Dynamic Sliding Window
Airborne Road-Marking Extraction
MMS Road-Marking Extraction
Target patch
Subdivide Each Survey into Patches
Airborne Road-Markings (Reference)
Window (Input)
MMS Patches
MMS Road-Markings
Gaussian Mixture Map Generation
Calculate the Transformation Matrix (NDT Matching)
MMS Calibration Using Transformation Matrixes
Precise Mobile Mapping Data

Figure 3-11. The overall workflow of the proposed 2D MMS registration framework.

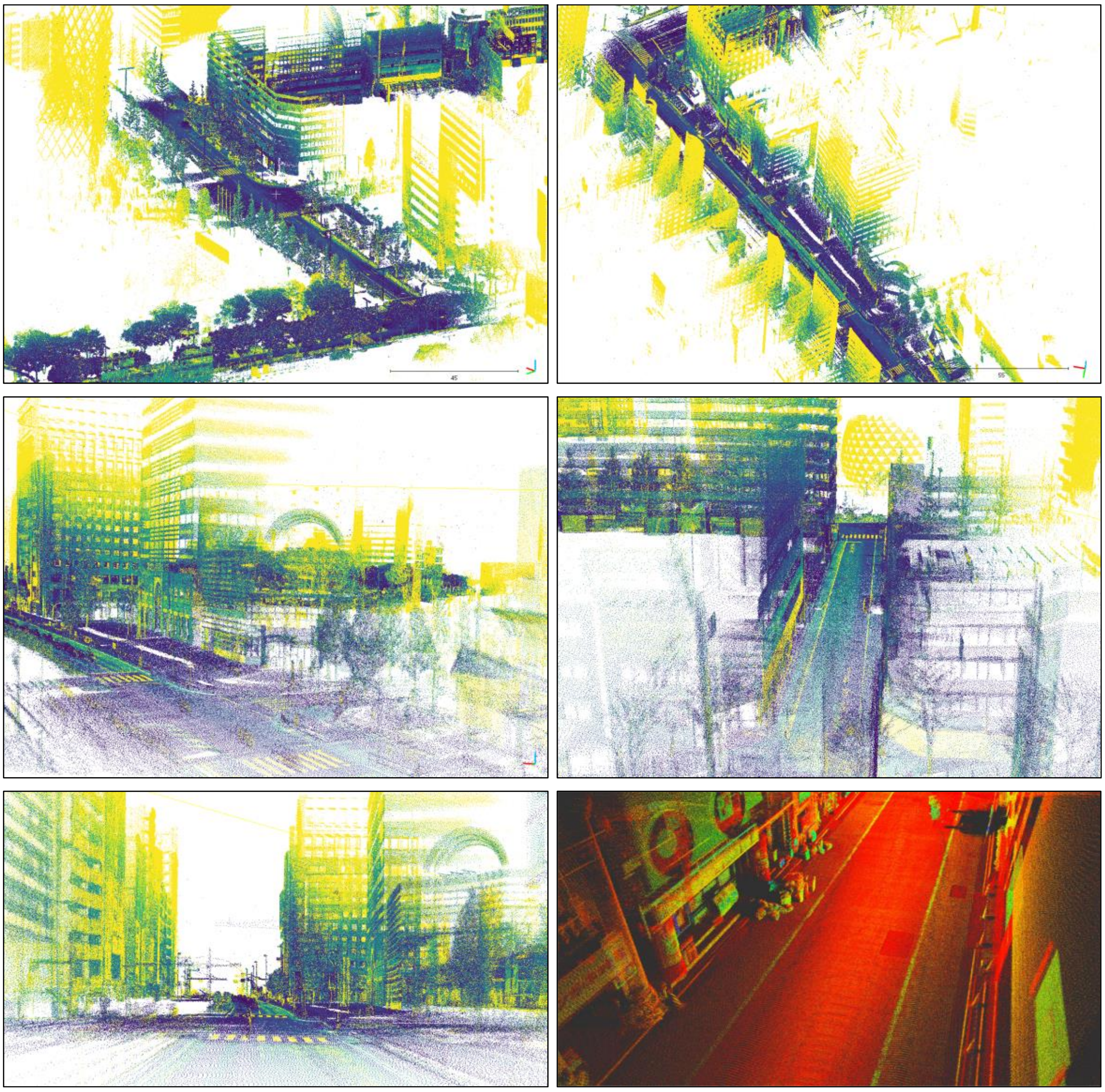


Figure 3-12. Generated map using the proposed mapping frameworks. The point cloud are subsampled by 10cm grids.

Chapter 4.
Map abstraction for Self-localization

Accurate vehicle self-localization is significant for the autonomous driving. Global Navigation Satellite System (GNSS) based localization techniques cannot achieve required accuracy in urban canyons. Simultaneous Localization and Mapping (SLAM) methods also suffer from error accumulation. 3D Light Detection and Ranging (LiDAR) has become popular recently. State-of-the-art localization approaches adopt LiDAR to observe the surrounding environment and match the observation with the priori known 3D point cloud map for understanding the position of the vehicle within the map. However, storing the massive point clouds needs an immense storage on the vehicle, and if they are stored on the servers, downloading them from the servers are again challenging. In this study, rather than using 3D point cloud directly as a map, we focus on the abstract map of buildings, which are mostly available in urban areas, easy to extract, and at the same time apparently observable by LiDAR. More specially, we proposed two methods to represent the abstract maps of urban areas. The first one is the multilayer 2D vector map of building footprints, which represents the building boundaries using vectors (lines). The second one is the planar surface map of buildings. Moreover, the two proposed abstract map share the same idea that the uncertainty (deviation) of each vector and the planar surface is calculated and included in the maps. Later in the localization phase, the observed data from LiDAR is matched with the prior map to obtain the precise location of the vehicle. Experiments conducted in one of the urban areas of Tokyo show that even though we extremely shrank the map size, we could preserve the mean error of the localization.

4.1. Introduction

The autonomous vehicles have considered as the key component of the intelligent transportation systems since their first introduction. In recent years, significant progress has been made in this field, and many companies already started their field tests. One of the critical requirements of the autonomous driving is the ability to know the ego-position (self-localization) within decimeters [1, 2]. Although the Global Navigation Satellite Systems (GNSSs) can achieve this level of accuracy in the open sky, its positioning quality degrades significantly in dense urban areas with tall buildings due to signal blockage and multipath effect [3-6]. One of the alternatives or auxiliary technology to GPS is a vision. Although vision-based methods have achieved a real progress [7-11], still this technology suffers from weather conditions, illumination changes, and shadows.

Recently, many companies are working on getting the LiDAR price as low as possible, and thus, this technology has once again gained a focus as the primary technology for the autonomous vehicle's perception. LiDAR's measurements are accurate, it has a wide field of view, long range, and it is irrelevant to different light conditions. Many companies have already started the miniaturizing and lowering the cost of this sensor to mount it on their commercial vehicles. In this chapter, we focus on the vehicle self-localization in urban areas using multi-channel LiDAR since, on the one hand, the GNSS cannot achieve the required positioning accuracy of the autonomous driving in urban areas and, on the other hand, plenty of features exist in the urban areas for the LiDAR localization.

LiDAR-based localization can be divided into map-based and SLAM-based (simultaneous localization and mapping) techniques. Methods without a prebuilt map are known as SLAM. These methods can be divided into two main categories: 1) feature-based [12], and 2) scan-based [13]. [14] uses building corners to rectify the error accumulation of the SLAM. In [15] author proposed a hybrid map-based SLAM using the particle filter. In their paper, the environment is represented by both grid and feature map. However, the typical methods based on the SLAM suffer from error accumulation [15].

With the development of remote sensing technologies such as LiDAR which is being used for airborne laser scanning (ALS) and mobile mapping system (MMS), the large-scale and high accuracy maps have become widely available [2]. The availability of the high-precision maps has increased the interests on the map-based vehicle self-localization techniques. In most of these methods, the high-definition 3D point cloud is used as the prior map, and in each sequence, the observed LiDAR scan is matched to the map to obtain the ego-vehicle position. In [1], the authors use the Velodyne laser scanner in addition to GPS, IMU, and odometer to generate the map offline. Then the localization within the map is performed using the particle filter. The authors extended their work in [16], developing a probabilistic approach. In [17], the authors develop an Iterative Closest Point (ICP) algorithm for local matching of the scans and incorporate a histogram-feature representation for the registration of sub-maps. In [18], lane markers are used as local features, which are extracted from reflectivity values of the LiDAR scans. In [19], Monte Carlo Localization (MCL) method utilizing the curb-intersection features is introduced, and the road observation is fused with odometry information. Pauly [20][64] proposed a height map encoding in point cloud by spectrally analyzing the point cloud.

After the ICP algorithm was proposed, Biber [21] introduced the idea of representing a two-dimensional environment by normal distributions for the scan registration. This method is known as Normal Distributions Transform (NDT). In the NDT, map space is divided into fixed 2D cells, and for each cell, a corresponding normal distribution is generated. Then, rather than matching the scan data to the point cloud, it is matched to the set of normal distributions. Later, Magnusson et al. extended the NDT idea to the 3D domain and proposed many variants for the NDT [22-25][65]–[69]. On the other hand, Tu et al. [26] was inspired by image compression techniques and proposed a method to compress the streaming point cloud using image and video compression methods. The authors also evaluated their compressed point cloud in the self-localization application. In our previous work [27], a multilayer 2D vector map was proposed for the vehicle self-localization which the map consisted of different heights of the building walls in 2D vector format. Therefore, the data size of the map was remarkably smaller than the conventional point cloud format. On the other hand, since the proposed vector map is multilayer and the normal distributions are generated from the vectors, we could achieve a better localization accuracy compared to the conventional methods based on 2D point cloud maps.

City scale 3D point cloud data is too bulky to be stored in the internal storages of the autonomous vehicles since each square kilometer consists of around 300 million points. For the self-localization applications, map structure should be small in size and, at the same time, capable of containing enough features for the localization. In the urban area, we have many structured objects such as buildings. These structures are made of many basic elements, such as 2D lines and 3D planes which are easy to extract and, at the same time, clearly observable by LiDAR. In this study, rather than using the massive 3D point cloud as the prior map, we employ two abstract map formats for the 2D and 3D localization of the autonomous vehicle: 1) multilayer 2D vector map, and 2) planar surface map. These maps are generated by extracting lines (vectors) and planes (planar surfaces) in addition to their uncertainties from the 3D point cloud. Therefore, we can abstract several thousands of points by single element, and as a result, we can extremely shrink the map size (25 million points to around 1000 elements). In the localization phase, the observed data form the LiDAR is registered into the prior map to obtain the precise location of the vehicle. Figure 4-1 shows the concept of the localization using the proposed abstract maps. The experimental result proves the effectiveness of the proposed conception of the abstract map-based localization.

4.2. Vehicle Self-Localization Based on Multilayer 2D Vector Map

4.2.1. Generating Multilayer Vector Map

Buildings are the most available and stable features in the cities, compared to trees, traffic signs, and poles. If the map is made based on buildings, updating and maintaining the map will be less challenging. On the other hand, LiDAR can observe building surfaces clearly. Even though the lower part of the buildings can be partially obscured by trees, cars or other dynamic objects but the upper layers of the building can be apparently captured by the LiDAR. Therefore, we only consider the buildings in the proposed map structures.

The 2D footprint of a building is generally considered as the area on a surface covered by the building at the ground level. However, as shown in Figure 4-2, a building can have different footprints in different heights. In this case, if a 2D map is generated based on only a specific height of the buildings (i.e. footprints), the self-localization of the vehicle might face a problem since the laser scanner installed on the vehicle might see a different height of the buildings which have different surface compared to the map. In this part, a vehicle self-localization method based on the multilayer 2D vector map is

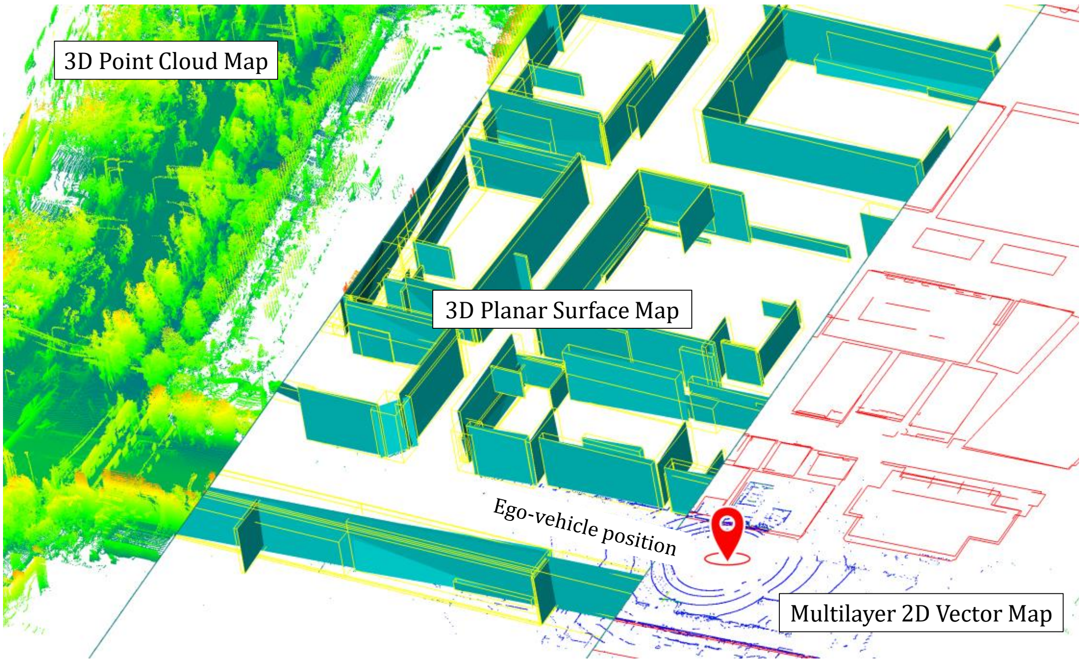


Figure 4-1. The concept of the vehicle self-localization using the proposed abstract map formats rather than massive 3D point cloud.

proposed to reduce the map size while maintaining the localization accuracy. In the proposed structure, instead of storing and employing massive 3D or 2D point clouds as prior map for the two-dimensional localization, 2D lines extracted from the 3D point cloud of the buildings are stored in the map as vectors. Thus, for each building, the footprints of the building at different height-levels are stored in the map, and as a result, a better localization can be expected. The term "multilayer" does not mean that a certain number of layers are considered in the map generation. In fact, it means that each building in the 2D vector map is generated from the multiple footprints if available (Figure 4-2).

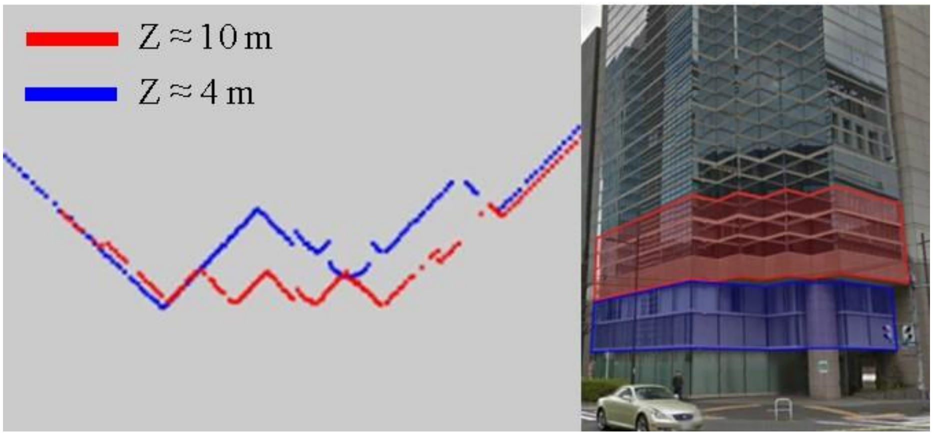
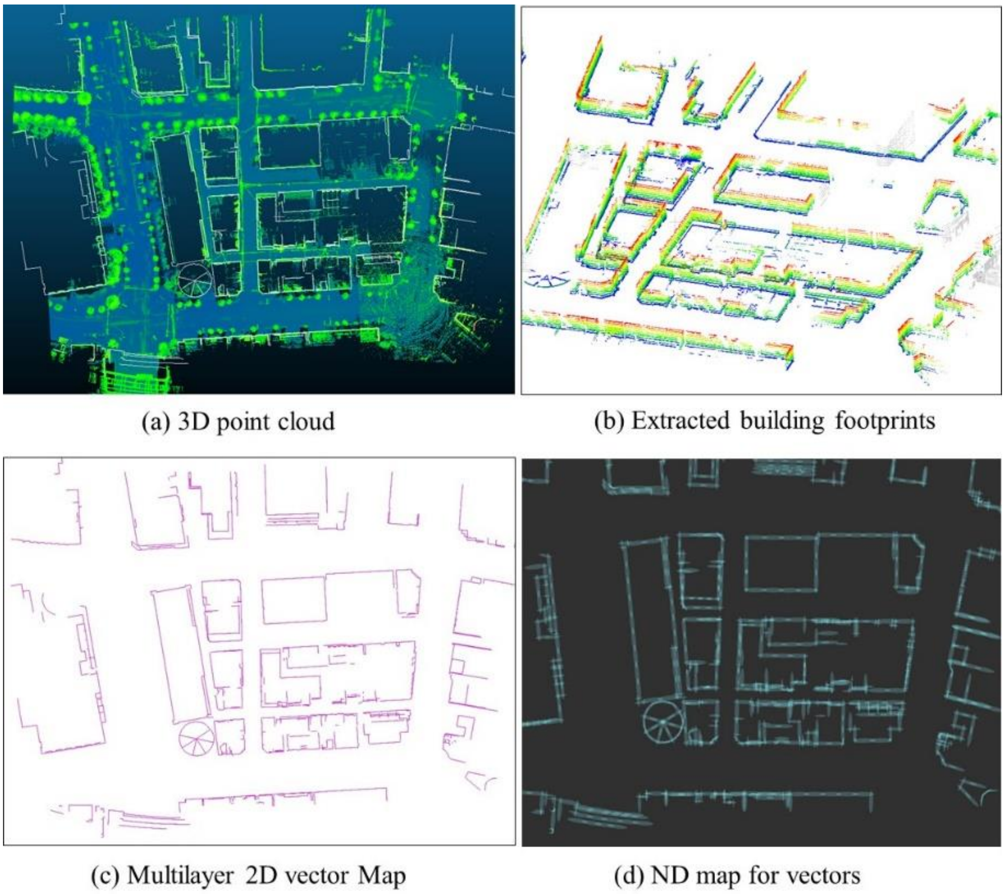


Figure 4-2. Building footprints for different layers (height *Z*) of the building.

To generate the multilayer vector map, the entire building points are extracted from the calibrated MMS point cloud (Figure 4-3 (a)) by referring to ALS, firstly. The obtained point cloud is shown in Figure 4-3 (b). Then, the remained points are projected onto the ground plane to form the multi-layer 2D point cloud which contains the building footprints of different heights. From this 2D point cloud, vector segments (lines) are extracted using the Random Sample Consensus (RANSAC). During the vector extraction, the uncertainty of each vector is also calculated and stored. Assuming that the length of an extracted vector $\vec{v}$ which represents a building wall is l, a 2D bounding box with the side lengths of l and $l/10$ is defined around the vector, as shown in Figure 4-4,. Given the points within this bounding box $\{\vec{y}_1, \dots, \vec{y}_n\}$, the uncertainty of the vector σ is calculated as

$$\sigma = \frac{1}{n}\sum_{k=1}^{n} Distance(\vec{v}, \vec{y}_k). \tag{1}$$


(a) 3D point cloud

(b) Extracted building footprints

(c) Multilayer 2D vector Map

(d) ND map for vectors

Figure 4-3. The procedure of generating the multilayer 2D vector map from the 3D point cloud. (a) Top view of the original 3D point cloud. (b) The building points extracted from the point cloud. (c) Multilayer 2D vector map. (d) Normal distributions of the vector map.

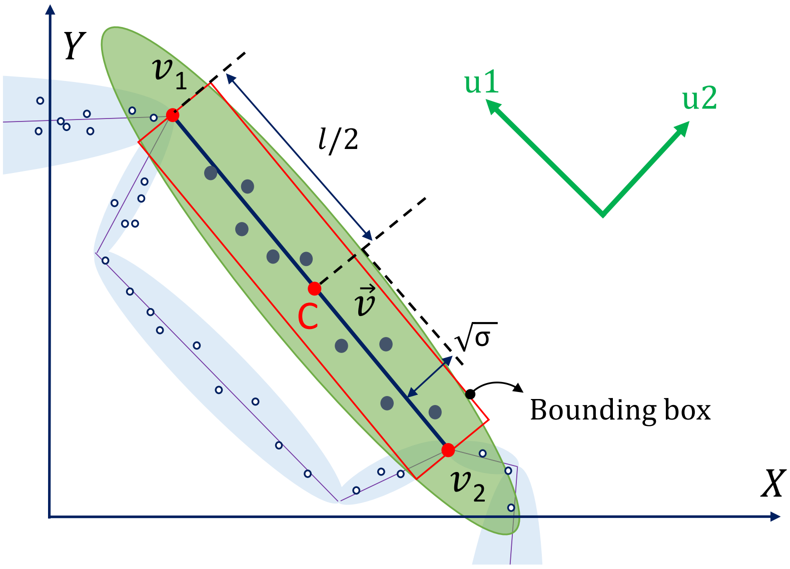


Figure 4-4. Calculating the vector uncertainty. The thick navy blue line shows the vector representing a building wall. The red box shows the bounding box to get uncertainty of vector representing the wall, and the green vectors are the Eigen vectors $\{u1, u2\}$ used for generating uncertainty for the vector. The green ellipse shows the normal distribution generated based on the center point of the vector and its uncertainty.

For each vector, the head and tail, and the uncertainty values are stored in the map structure. Figure 4-3 (c) shows the multilayer 2D vector map generated for the selected area, and Figure 4-3 (d) visualizes the vector uncertainties by the blue ellipses.

4.2.2. Scan-to-Vector-Map Matching for Localization

To estimate the current state of the vehicle, the laser scan should be registered to the prior abstract map. The flowchart of the proposed localization method is shown in Figure 4-5. Firstly, the vector map is converted to a normal distribution (ND) map, and the laser scan is preprocessed before the matching. In the map matching phase, the laser is matched to the ND map using the point-to-distribution variant of the NDT (P2D-NDT).

4.2.2.1 Generating Normal Distribution (ND) from 2D Vector Map

To perform the fast and robust matching between the LiDAR scan and 2D vector map, the NDT-matching is used. In the NDT-based point cloud registration, unlike the iterative closest point (ICP), the reference point cloud is converted to a set of Gaussian distributions, and then the matching is performed over these distributions [24]. In the conventional NDT, to calculate a set of NDs which represents the map, a process called discretization is performed which divides the map space into grids with a fixed size.

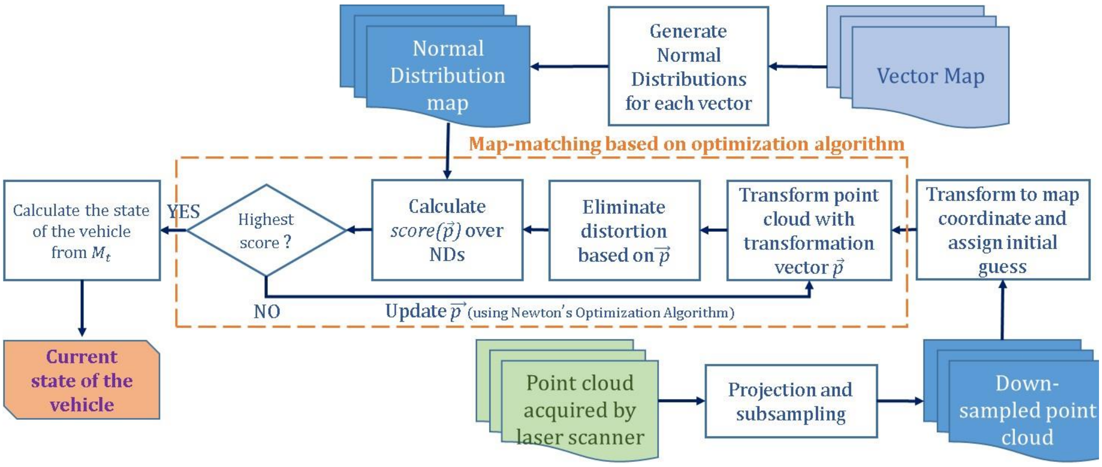

Normal Distribution map
Generate Normal Distributions for each vector
Vector Map
Map-matching based on optimization algorithm
Calculate the state of the vehicle from M_t
YES
Highest score ?
Calculate $score(\vec{p})$ over NDs
Eliminate distortion based on $\vec{p}$
Transform point cloud with transformation vector $\vec{p}$
Transform to map coordinate and assign initial guess
NO
Update $\vec{p}$ (using Newton's Optimization Algorithm)
Current state of the vehicle
Point cloud acquired by laser scanner
Projection and subsampling
Down-sampled point cloud

Figure 4-5. Flowchart of the vector map-based self-localization. Inputs are the vector map and point cloud acquired by the laser scanner, and the output is the current position of the

Usually, the uncertainties in the reference ND map come from this discretization process which results in a considerable alignment error. Finding the suitable grid size is a difficult task and depends on many parameters such as the environment, sensors, and etc. The larger the grid size becomes; the more uncertainty appears in the NDs. If a small grid size is chosen to limit the uncertainty of the NDs, then a considerable number of the grids will not have enough point for making ND (5 points), and therefore we will have a sparse map without enough features which is not suitable for the localization. Moreover, in the conventional NDT method, two close walls might fall into a single cell of the grid, and therefore abstracted by one ND. To avoid the problems mentioned and maintain the localization accuracy, this paper generates the NDs based on the extracted vectors and their uncertainty values.

To generate the ND from the 2D vectors in the map, we need to define the mean and covariance for each of them. The probability of observing a point p in $\mathbb{R}^2$ on a particular vector represented by a normal distribution is derived from the following equation:

$$P(p) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp(-\frac{(p-\vec{\mu})^T\Sigma^{-1}(p-\vec{\mu})}{2}) \tag{2}$$

where $\sum$ is the covariance matrix, and $\vec{\mu}$ is the mean. In the case of vectors, their center points (Figure 4-4) are selected as the mean values. To define the covariance matrices, Eigen decomposition is used. The covariance matrix for the vector $\vec{v}$ can be defined as:

$$\Sigma_v = \mathrm{U}\Lambda U^T = \begin{bmatrix} u1_x & u2_x \\ u1_y & u2_y \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} u1_x & u1_y \\ u2_x & u2_y \end{bmatrix}, \tag{3}$$

where *u1* and *u2* are the Eigen vectors, and λ_1 and $\lambda_2$are the Eigen values. The Eigen vectors are obtained from the head and tail points of the vector segments as follows:

$$u1 = \begin{bmatrix} v_{1_x} - v_{2_x} \\ v_{1_y} - v_{2_y} \end{bmatrix},\ u2 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} u1, \tag{4}$$

where v_1 and v_2 are the head and tail points of the vector $\vec{v}$. λ_1 is defined as half of the length of the vector, and λ_2 is set to the uncertainty of the vector (σ) which was calculated in Eq. (1). The NDs generated from the vectors are shown in Figure 4-6. As each vector has its own ND, the uncertainties caused by the discretization in the conventional NDT can be resolved, and therefore, building corners and neighboring walls can be represented correctly.

4.2.2.2 Scan to Normal Distribution (ND) Vector Map Matching

After obtaining the ND map from the multilayer 2D vector map, we have to localize the vehicle within the map. For the localization, a multi-layer (channel) LiDAR mounted on the roof of the vehicle is used. Usually, the distortion of the LiDAR scan which is caused by the vehicle motion during the experiment is eliminated using a relative movement acquired from odometer sensors. In our algorithm, we proposed a method to remove the distortion inside the optimization process of the map matching. In this case, as can be seen in the flowchart of the map matching in Figure 4-5, each time the transformation vector transforms the scan, the distortion elimination process is performed to calculate the distortion based on the newly estimated position.

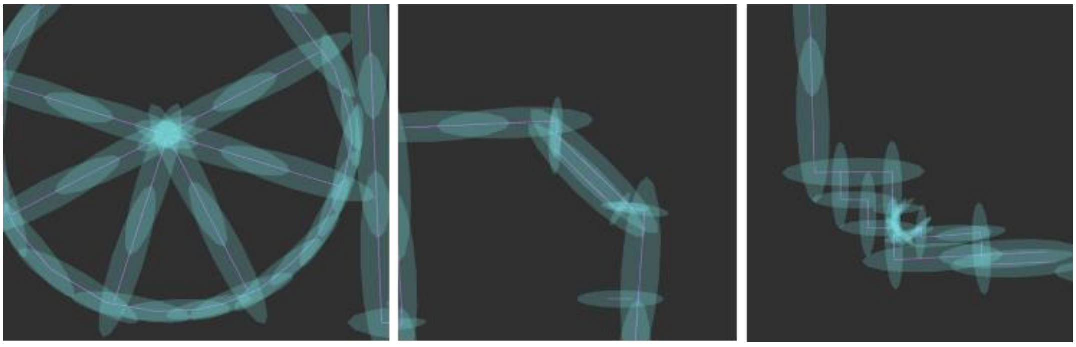


Figure 4-6. Normal Distributions generated from the vectors.

To perform a 2D matching, the multi-layer laser scan should be converted to a 2D point cloud by projecting all layers to the ground plane. Employing a multi-layer laser scanner instead of a single-layer one brings two advantages for the map matching. Firstly, it increases the point density of the building footprints. And secondly, scanning different levels of buildings allows us to capture different footprints of a single building. Having access to a dense scan makes the registration robust against outliers. To avoid bias from uneven point distribution and to reduce the matching time, the input scan should be down-sampled [23]. In the proposed method, to keep the benefits of the higher point density around building walls, the down-sampling is done before the 2D projection. For the down-sampling, we create a 2D grid of $30\,\text{cm} \times 30\,\text{cm}$ over the input scan. And then, all points within each grid will be approximated with their centroid.

After preprocessing mentioned above, the 2D scan is transformed from the vehicle coordinate to the map coordinate (Global) with an initial guess acquired from two previous subsequent positions. Then, the best alignment is obtained which is the transformation matrix M_t. In fact, M_t is the optimal 2D transformation matrix that applied to initially aligned laser scan to further match the scan to the ND map, and defined as

$$M_t = \begin{bmatrix} R_z(\theta) & T_{x,y} \\ 0 & 1 \end{bmatrix} \tag{5}$$

where $R_z(\theta)$ describes θ degrees of the rotation around the Z-axis, and $T_{x,y}$ is the 2D translation. This optimal matrix is obtained by maximum likelihood estimation [27][70].

Suppose that $X = \{x_1, \dots\ , x_n\}$ is the initially transformed input scan in $\mathbb{R}^2$, and P_j is the ND for the vector segment j. The optimal M_t is the transformation that maximizes the likelihood function (score function)

$$\text{Ł} = \prod_{k=1}^{n} P_j(M_t x_k) \tag{6}$$

or, equivalently, minimizes the log-likelihood function of Ł:

$$-\log \text{Ł} = -\sum_{k=1}^{n} \log P_j(M_t x_k) \tag{7}$$

This cost function is less complex for calculating gradient and Hessian matrix in optimization process thus, Eq. (7) is used for obtaining M_t. In Eq. (7), outlier points significantly decreases the matching score, thus instead of the normal distribution, a mixture of uniform and normal distribution $\acute{P}_j$ is used:

$$\acute{P}_j = \xi_1 P_j + \xi_2 P_0 \tag{8}$$

where P_0 is the expected rate of the outliers, and ξ_1 and ξ_2 are constants such that Eq. (8) integrates over the j_{th} vector to one. By applying Eq. (8) to the log-likelihood function in Eq. (7) and approximating it for the sake of simplicity, the score of k_{th} point can be defined as:

$$Score(M_t, x_k, j) = d_1 \exp(-d_2 \frac{(M_t x_k - C_j)^T {\sum_j}^{-1} (M_t x_k - C_j)}{2}) \tag{9}$$

where j is index of the closest vector to the transformed point $M_t x_k$, $\sum_j$ and C_j are covariance and center of the j_{th} vector, and d_1 and d_2 is obtained from ξ_1 and ξ_2. Finally, the cost function of the transformation matrix M_t for the input scan X is defined as:

$$Cost(M_t) = -\sum_{k=1}^{n} Score(M_t, x_k, j) \tag{10}$$

To find the optimal M_t, the Newton optimization method is employed. Newton's method optimizes the cost function by

$$H \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix} = -g, \tag{11}$$

where H and g are the Hessain matrix and the gradient of the cost function, and x, y, and θ are the translation and rotation parameters of the Matrix M_t. After the Newton's process is merged, the optimum M_t is obtained and the ego-position of the vehicle can calculated by multiplying M_t to the initial translation matrix. Figure 4-7 shows a few examples of the proposed map matching.

4.3. Vehicle Self-Localization Based on Probabilistic Planar Surface Map

The vector map format proposed in the previous chapter is suitable for the 2D localization of the autonomous vehicle, but cannot be used for the 3D localization.

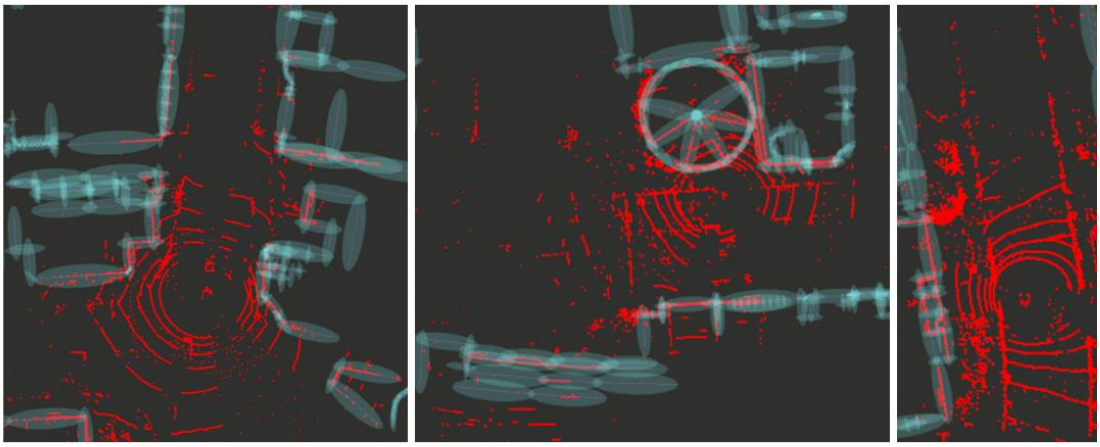


Figure 4-7. A few examples of the localization technique based on the multi-layer 2D vector map. The input laser scans are shown in red. The blue spheres are the NDs generated from the vector map.

Besides vectors, planes can be considered as the abstract three-dimensional representation of building walls. Thus, we extend the idea from Section 2 and propose a 3D planar surface map format for the vehicle localization. The flowchart of the proposed planar surface map-based localization is shown in Figure 4-8. As can be seen in the figure, the output of the planar surface map-based localization is the 6D state of the vehicle.

4.3.1. Generating Probabilistic Planar Surface Map

In contrast to the 2D vector map-based localization proposed in the previous chapter which the ground points were removed in the vector mapping step, the ground surface plays an essential role in the 3D localization using the planar surface map, especially for defining the pitch and height of the vehicle position. Therefore, instead of removing the ground surfaces, we actively take them into account for the localization. Thus, the primary components representing a planar surface map will be the building walls and ground surface.

The first step of creating a planar surface map is to define the ground surface from the 3D point could map (Figure 4-9 (a)). The ground points are obtained using the cloth simulation filter algorithm proposed in [29][71] which works well for large-scale ground detection. Figure 4-9 (b) and (c) show the ground extraction procedure. Typically, the ground surface does not include so many details and also does not change so often. Therefore, a local surface of the ground can be abstracted by a plane without losing

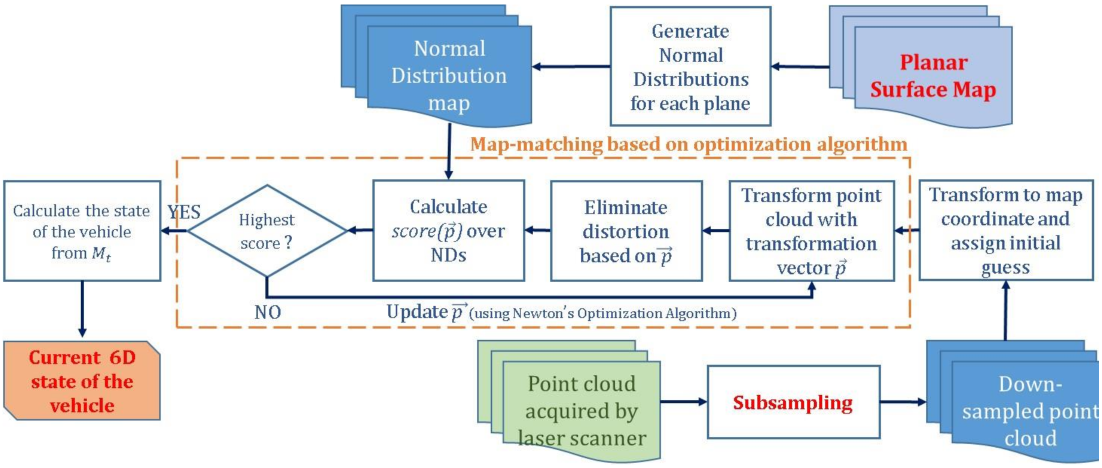

Normal Distribution map
Generate Normal Distributions for each plane
Planar Surface Map
Map-matching based on optimization algorithm
Calculate the state of the vehicle from M_t
YES
Highest score ?
Calculate $score(\vec{p})$ over NDs
Eliminate distortion based on $\vec{p}$
Transform point cloud with transformation vector $\vec{p}$
Transform to map coordinate and assign initial guess
NO
Update $\vec{p}$ (using Newton's Optimization Algorithm)
Current 6D state of the vehicle
Point cloud acquired by laser scanner
Subsampling
Down-sampled point cloud

Figure 4-8. Flowchart of the planar surface map-based vehicle self-localization. Inputs: the planar surface map and 3D laser scan. Output: the 6D state of the vehicle.

information. To generate the ground map, the ground surface is first divided into large grids (10 m × 10m), and then the plane detection algorithm is applied to points within each cell. For each detected plane, the uncertainty of the points forming the plane is also estimated and stored. Therefore, for each ground surface the center, width, length, and normal of the plane are stored beside the mean and sigma of the corresponding distribution. The method of calculating the ND for each surface is described after the building surface extraction for both the ground and building surfaces. Figure 4-9 (d) shows the normal distributions of the ground surface.

After the ground extraction fr om the MMS point cloud, remaining points are used to extract the building walls, as shown in Figure 4-9 (e). The objective of the plane extraction is to find the main features that likely to be observed by the laser scanner in the localization phase and not 3D building reconstruction. Planar surfaces are extracted by efficiently fitting the planes to the off-ground points using the RANSAC [30][72]. RANSAC randomly generates plane candidates in each iteration by subsampling the point cloud and estimates the plane with maximal score. The plane is only accepted if the deviation of the points is less than a predefined threshold. We empirically defined the angular threshold for extracting the planes within the experimental area which is three degree. The remaining RANSAC parameters used for the plane extraction are as follows; maximum distance to the plane is set to 1.0 m, the minimum number of points per each plane is set to 200, and the overlooking probability is set to 0.01. The results of plane

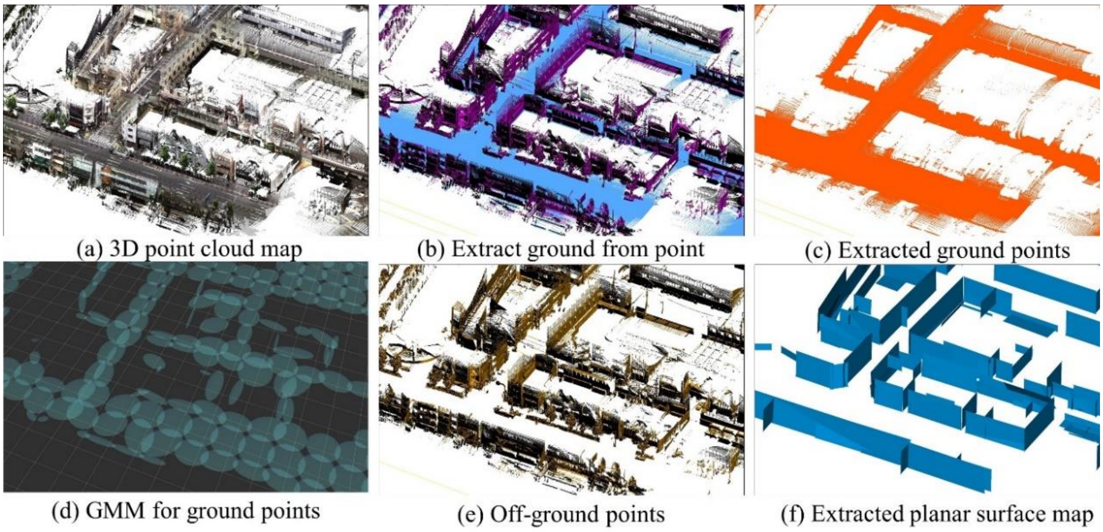

(a) 3D point cloud map
(b) Extract ground from point
(c) Extracted ground points
(d) GMM for ground points
(e) Off-ground points
(f) Extracted planar surface map

Figure 4-9. The procedure of generating the planar surface map. (a) The original 3D point cloud. (b) The ground segmentation result. (c) The extracted ground. (d) The Gaussian mixture model of the ground surface. (e) The off-ground points. (f) The planar surface map of the off-ground points.

fitting are shown in Figure 4-9 (f), where each plane represents the building walls. For each extracted building plane, just like the ground part, the center, width, length, and normal of the plane, are stored with the mean and covariance of the corresponding ND of the plane.

As mentioned before, to use the planar surfaces extracted from both the ground and building surfaces for the self-localization, we have to generate NDs which will be used in the scan matching phase. Therefore, the uncertainty of the points forming each plane should be calculated and stored for making the ND for the plane.

Suppose that a set of points $\mathrm{Y} = \{y_1, y_2 \ldots, y_n\}$ in $\mathbb{R}^3$ is forming the $\mathrm{i_{th}}$ planar surface entity P_i. The uncertainty of the plane P_i is defined by the mean μ and covariance matrix $\sum$ as follows:

$$\mu = \frac{1}{n}\sum_{k=1}^{n} y_k \tag{12}$$

$$\sum = \frac{1}{n-1}BB^T,\ B = [y_1 - \mu, \ldots, y_n - \mu\,] \tag{13}$$

By having the mean and covariance matrix, the probability of observing a point p in $\mathbb{R}^3$on a particular wall surface or ground surface represented by a normal distribution with the estimated covariance and mean is derived from the following equation:

$$P(p) = \frac{\exp(-\frac{1}{2}(p-\mu)^{\mathrm{T}}\Sigma^{-1}(p-\mu))}{\sqrt{(2\pi)^3|\Sigma|}} \tag{14}$$

In the planar surface map, we only store the planes and their uncertainty values for both the ground and building surfaces, and neglect all other details to reduce the map size. While those details might be helpful for the self-positioning, especially in the longitudinal direction, the acquired surface information can provide fairly enough features for the map matching in the urban areas. Figure 4-10 shows different map formats and abstraction level of a particular urban area. As can be seen, the dense 3D point cloud consists of higher details which lead to a massive data size. In the conventional NDT, the grid size and localization accuracy are tightly connected. While increasing the grid size can reduce the final map size, the positioning accuracy is affected and degraded. The proposed 3D surface map can provide the smallest map size and comparable localization accuracy with the original point cloud.

4.3.2. Scan-To- Planar -Map Matching for Localization

4.3.2.1 Making Normal Distribution (ND) from Planar Surface Map

In the previous step, the planar surface map of the environment including the ground and building surfaces was generated and stored. In this step, the planar surface

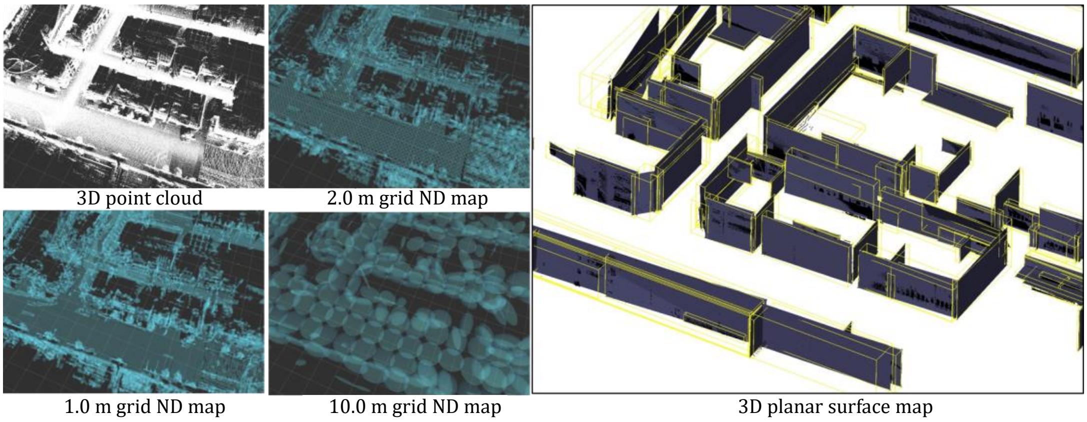


Figure 4-10. Comparison of different map formats and abstraction levels: point cloud map, fixed-size grid ND map, and 3D planar surface map.

map should be converted to the ND map to perform the NDT matching for the localization. If we directly use the uncertainty associated with a surface in the planar surface map as the ND (Eq. 14) and perform the NDT matching, we will face two challenges as written below.

In the NDT matching technique, to calculate the matching score, we should define the corresponding ND for each point in the input scan. This is done by searching and selecting the ND with the nearest centroid to the point. This method works well if the domain of NDs is relatively small. However, in the planar surface map, typically the NDs representing the planes are large. Therefore, the distance to the ND centroids is not a good estimate for defining the nearest ND, as shown in Figure 4-11 (a). In the figure, the closest surface to the point C is P_1. However, as the centroid of P_2 is closer to C than the centroid of P_1, P_2 is considered as the corresponding ND for the C. The same problem happens for points A and B, and it causes a significant matching error. To solve this, point to plane distance should be considered instead. However, calculating the distance of all points in the input scan to all planes in the map is time-consuming and makes the matching slow.

The second challenge is the score calculation. If each plane is represented by only one ND, the input points with a same distance to the plane might get totally different scores. In Figure 4-11 (a), points A and D have an almost same distance to plane, but the score of A is much lower than D.

To overcome the problems mentioned above without increasing the size of the planar surface map, each plane is subdivided to a fixed smaller size and then represented by multiple NDs in the localization phase, as shown in Figure 4-11 (b). By doing this, the nearest ND to the input points can be estimated by searching the centroids of the NDs by k-d-tree nearest neighbor algorithm which is very fast, and also the score function will become more uniform.

4.3.2.2 Scan to Normal Distribution (ND) Plane Map Matching

Before the map matching, the sensor data collected from the laser scanner is preprocessed for removing the distortion using the method described in section 2.2-B. After excluding the distortion, scans are down sampled using the similar method in 2.2-B. Then, the scan is transformed from the vehicle coordinate to the map coordinate by an initial guess. The initial guess of the first frame is obtained from GPS, but after that, it is

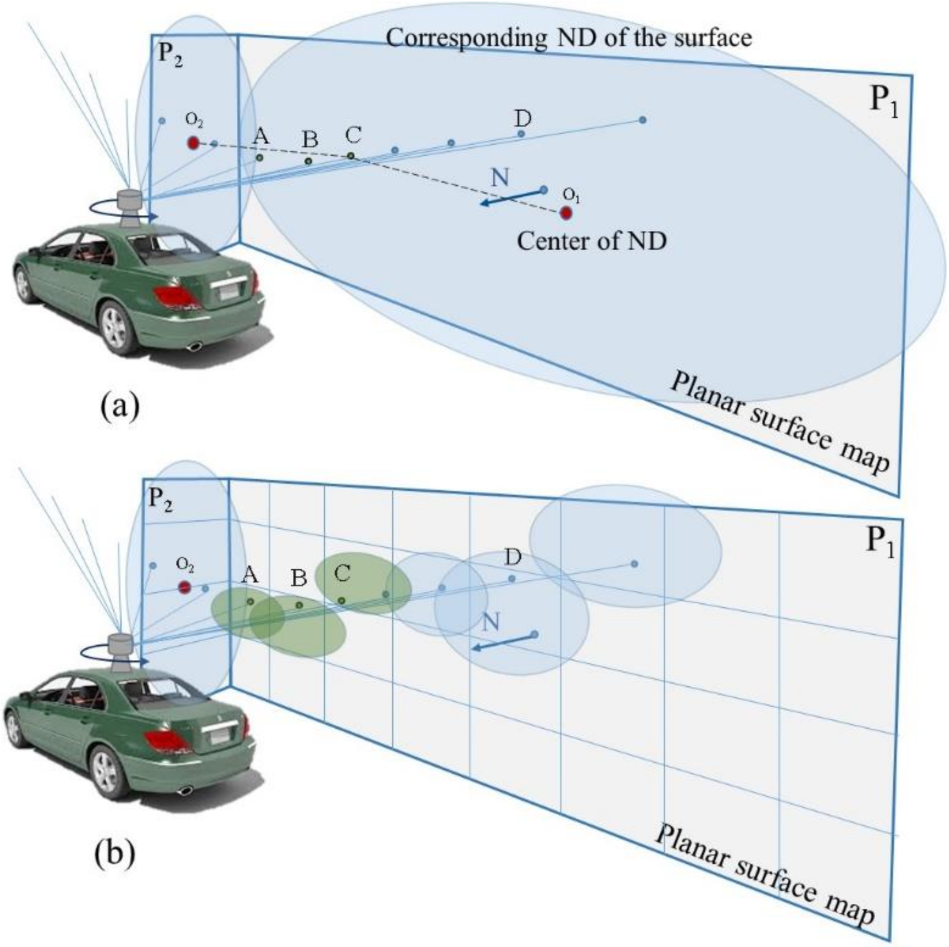


Figure 4-11. Effect of the NDs generated from the planar surface map on the defining the correct corresponding ND of the points in the input scan. (a) Misdetection of the nearest ND due to the large ND domain, and the score reduction due to the distance of the input points from the center of the ND. (b) Effect of representing the planar surface using smaller NDs.

estimated by a simple prediction performed based on two previous positions of the vehicle.

Suppose that the transformation vector for the initial guess is $\vec{p} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z,]^T$ where t_x, t_y, t_z are the translation, and $\theta_x, \theta_y, \theta_z$ are the rotation parameters. Assuming that $X = \{x_1 \dots\ x_m\}$ is the subsampled laser scan and X' is the scan after applying the transformation $\vec{p}$. For each transformation $\vec{p}$, the motion of the vehicle is estimated and the distortion is removed based on the estimated motion to make a distortion free scan $\hat{X}$. Similar to section 2.2-B, the cost function of $\hat{X}$ is defined as

$$Cost(\hat{X}) = -\sum_{k=1}^{n} Score(\hat{X}) \qquad (15)$$

Assuming that T is the transformation function and D is the distortion elimination function, $\hat{X}$ can be defined as

$$X' = T(\vec{p}, X) \qquad , \quad \hat{X} = D(X') = D(T(\vec{p}, X)) \tag{16}$$

Therefore, Eq. (15) can be expanded as:

$$Cost(\vec{p}) = -\sum_{k=1}^{n} Score(D(T(\vec{p}, X))) \tag{17}$$

To find the optimal $\vec{p}$ which is the best matching result, the Newton optimization method is applied. Newton's method optimizes the cost function by iteratively changing the $\vec{p}$ using the following equation:

$$H[\Delta x \quad \Delta y \quad \Delta z \quad \Delta\theta_x \quad \Delta\theta_y \quad \Delta\theta_z]^T = -g \tag{18}$$

where H and g are the Hessain matrix and the gradient of the cost function, and x, y, z are the translation and $\theta_x, \theta_y, \theta_z$ are the rotation parameters of $\overrightarrow{p}$. After the Newton's process is converged, the ego-position of the vehicle can be calculated using the optimum $\overrightarrow{p}$. For the details of the optimization process, the reader is referred to [28][70].

4.4. Experimental Results

This section presents the experimental results of the proposed map structures and localization method. To evaluate the performance of the localization method using the proposed map formats, the experiments were conducted in the vicinity of Hitotsubashi, a typical urban area in the Chiyoda-ku area of central Tokyo, Japan. Figure 4-12 shows route of the experiment with the length of 650 m and accuracy of the GPS-based localization in the area. For the experiment, we run the vehicle four times in the same route and compared the 2D and 3D localization accuracy of different map formats.

Figure 4-13 (left) shows the MMS used for the mapping and Figure 4-13 (right) shows the experimental vehicle and the sensor setup for the localization. The 3D point cloud map is obtained by two single layer SICK laser scanners, and offline calibration using our previous work [2, 31]. For the localization, our experimental vehicle is equipped with Velodyne VLP-16 laser scanner with 16 channels which has 100m range, 360-degree horizontal and 30-degree vertical field of view. The laser scanner is installed at a height of 2.3 meters and set to spin at 10Hz which collects ten scans in each second. The maximum speed of the vehicle in the experiments was 47 Km/h. Since defining the

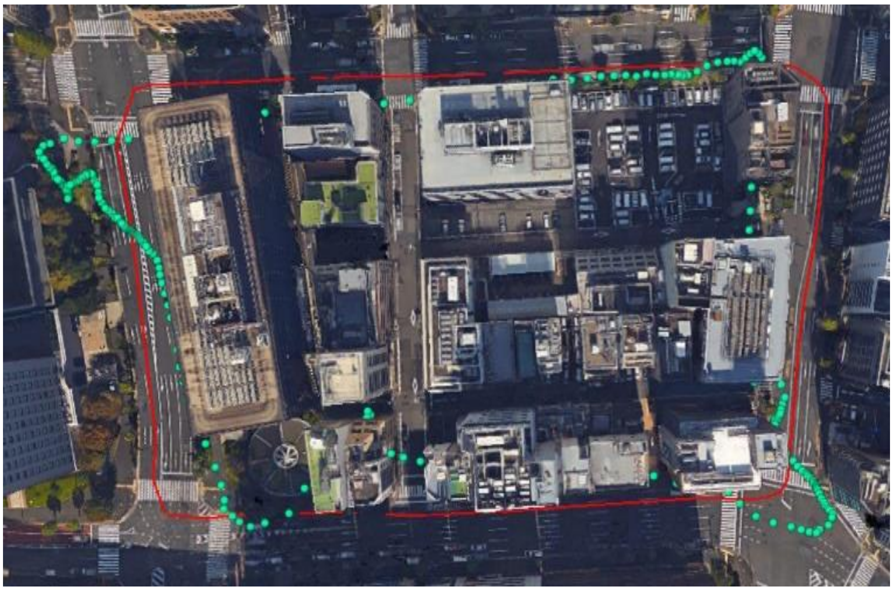


Figure 4-12. The experimental area in the vicinity of Hitotsubashi, a typical urban area in the Chiyoda-ku area of central Tokyo, Japan. The red line shows the experiment route and the green dots show the positon acquired by GPS for the route.

ground truth in this route is challenging, we adopted the 3D point cloud-based localization method as a quasi-ground truth.

4.4.1. Evaluation for Multiplayer Vector Map-based Localization

There are three aspects needs to be evaluated. First, the proposed method can

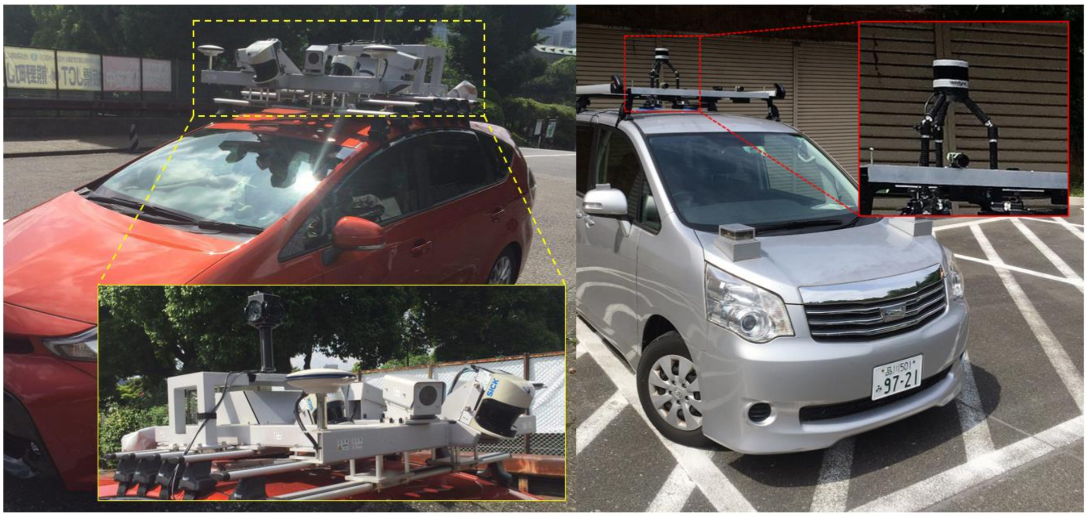
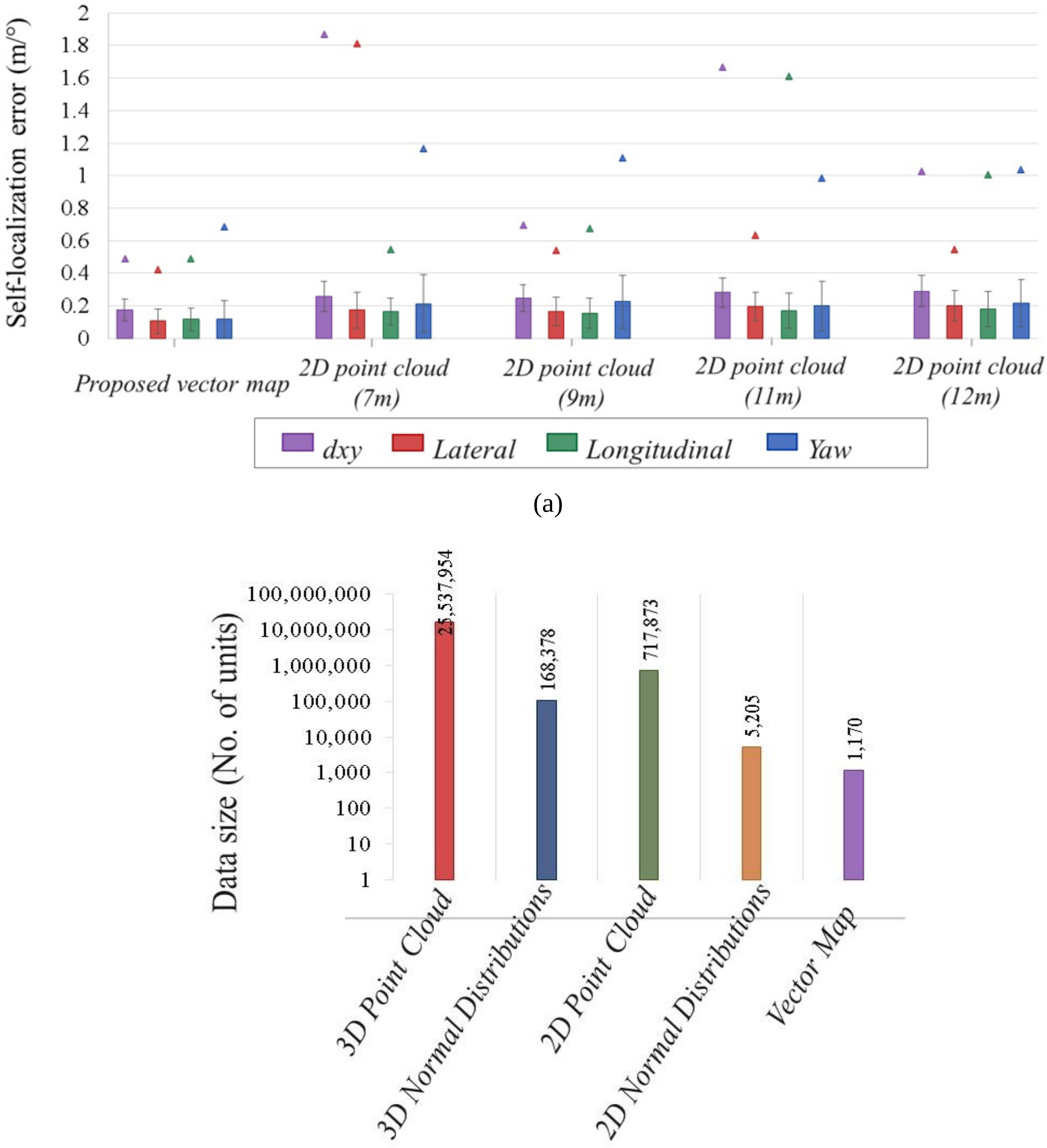


Figure 4-13. The experimental setup. The left is the MMS used for the mapping and the right is our experimental vehicle. Velodyne VLP-16 is mounted on the top of this car at a height of 2.3m.



(a)

(b)

Figure 4-14. First chart of (a) shows the evaluation of proposed method by comparing to ground truth, and as result shows, the mean error is less than 20 cm. Thus, the proposed method extremely decreased the map size while preserving the localization accuracy. Other chart of (a) show the comparison of proposed method with 2D point cloud method. Figure (b) shows the number of map elements in each map structure.

preserve the localization accuracy comparing to the ground truth, while our method uses an extremely small map structure. Second, the proposed method can outperform 2D point cloud methods in terms of accuracy and finally, show the effect of vector base discretization in the localization accuracy.

Figure 4-14 (b) shows the number of elements in each map structures. The proposed

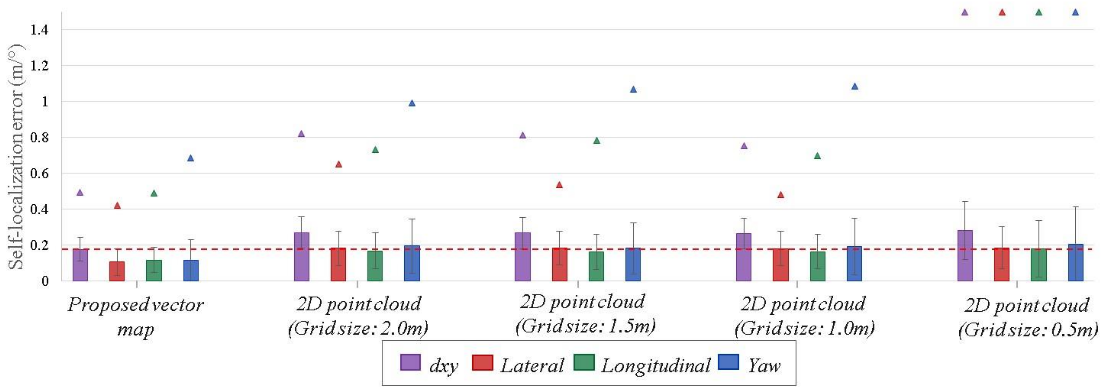


Figure 4-15. Comparison of proposed method with 2D point cloud with different grid size. Height of building footprint used in 2D point cloud methods is 6m. As results show proposed method perform better comparing to all grid sizes.

method is the lightest structure comparing to all of the other methods. Figure 4-14 (a) shows the self-localization results of proposed method and it's comparison to 2D point cloud methods. In this figure, mean, maximum and variance of self-localization errors are shown. As can be seen in the first chart of Figure 4-14, the average self-localization error of the proposed vector map is kept less than 20cm which shows, though we extremely lowered the map size, we could fairly preserve the localization accuracy. Rest of the charts in Figure 4-14 shows the comparison with 2D point cloud based methods. These methods are based on single layer map (certain height of buildings footprints) and use multilayer laser scanner for getting input scans. For NDT discretization, they used 1m grid size. The downsampling strategy is same in all methods. As we can see in Figure 4-14, our proposed method outperforms single layer 2D methods in terms of mean and max error. In Figure 4-16, 2D point cloud map is made from all layers to eliminate the effect of different layers in localization so that we can compare vector based discretization and static grid based discretization. Our method still has better performance. Figure 4-15 shows the comparison of proposed method with the different discretization of the 2D point cloud. As shown in Figure 4-15, the vector map outperforms the 2D point cloud based method with different grid size as well.

4.4.2. Evaluation for Plane Surface Map based Localization

Figure 4-17 shows the comparison of plane surface map size for the experimental area. A total number of point cloud data for this area is around 25 million points. Grid representation of the map can decrease the map size. For example for 1.0m grid size, a

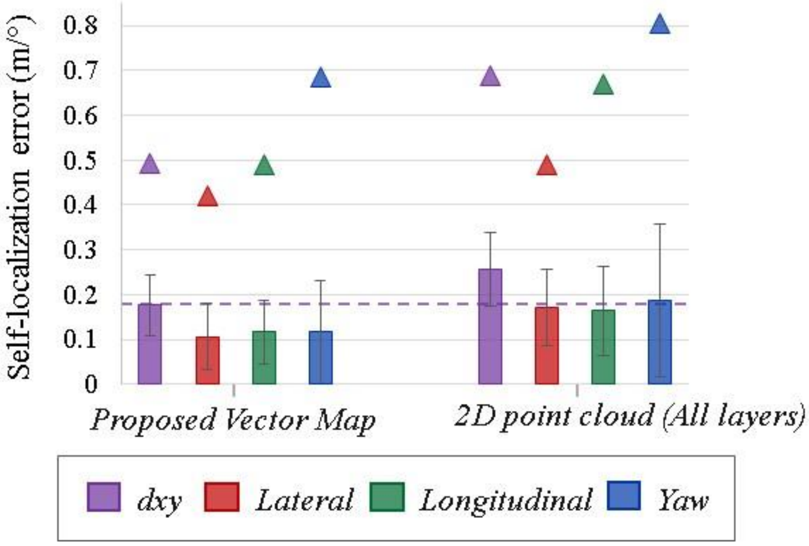


Figure 4-16. Comparison between proposed method and 2D point cloud which uses all footprints in different heights.

total number of the map entity for the same area become 168,378. For the same area, a total number of extracted planar surface entity is 1,043 which is extremely lighter than grid-based and point cloud based representation.

Figure 4-18 shows the accuracy of proposed plane surface map based localization. As Figure 4-18 shows mean error of proposed method comparing to the ground truth is 43cm and maximum error is 120cm. These results demonstrate that, though we used an extremely lighter map structure, we could preserve the localization accuracy. Figure 4-18 also illustrates the accuracy of proposed method comparing to NDT-based methods. As can be seen Figure 4-18, as grid size of NDT-based methods increase the accuracy

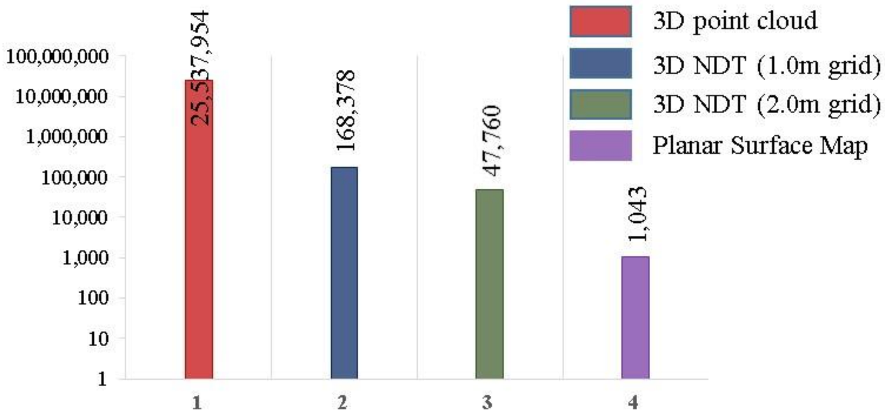


Figure 4-17. Map size comparison of proposed plane surface map and other point cloud based methods. A base-10 log scale is used for the Y axis. As can be seen in this figure, planar surface map structure is extremely lighter than others.

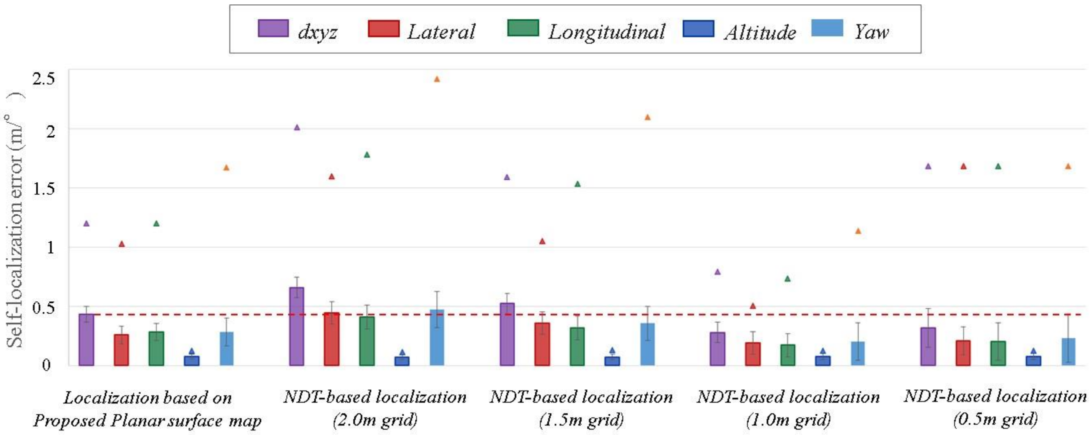


Figure 4-18. Accuracy of the plane surface based self-localization and other NDT-based methods.

decrease. This is because more abstraction applied to the map and thus many details of the map is removed. Also, in higher grid size, the edge of the buildings which vital for accurate map matching are also removed. Proposed method has lower error comparing to 2.0 NDT-based localization while map size is extremely lighter. As Figure 4-18 shows, best discretization size for the NDT-based method in this area is 1.0m. In NDT-based localization methods, the optimum size of grids depends on the environment and there is no method for finding optimum grid size for each environment. As can be seen from Figure 4-18, in proposed method, the smallest error is for altitude. This is because the ground does not have so many details and can be abstracted without any loss. Also distortion does not have so much effect on altitude. The evaluation results of Figure 4-18 shows that in proposed method, the longitudinal error is more than lateral error. This is because in some part of the experimental route, buildings are placed very densely, and the space between two buildings are so minor. Therefore, the lateral walls (walls which are perpendicular to the moving direction of the vehicle) cannot be fully captured by the MMS in mapping phase. As a result, points for such a wall become very few, and that part of the building cannot be extracted in the planar surface map. Therefore, lack of those lateral features results in error in positioning. This can be solved by manipulation of the plane extraction parameters for such position. One possible future work for proposed method is to find out such position and add more details to the map in order to get better longitudinal accuracy.

Chapter 5.
Quantification of digital map ability for self-localization

5.1. Introduction

In map-based localization methods, the source of localization error can be divided into three main categories. These are input scan, matching algorithm, dynamic phenomena, and map. In this chapter, first each of the source of error are explained

- **Input scan**

  - **Input scan quality**

    Input scan quality can change the localization accuracy. For obtaining input scan, laser scanner is used. Type of laser scanner is important to get more detailed scan of the environment. For example Velodyne HDL-64 has 64 channels (layers) of laser beams which can capture the 360˚ environment densely. Velodyne VLP-32C has 32 channels and VLP-16 has 16 channels. These three laser scanners capture the environments with different quality. HDL-64 can capture the details of surrounding which cannot be captured by VLP-16. These details can help the matching algorithm to match the input and map more accurately. In addition to number of channels, other sensor related parameters such as vertical and horizontal field of view, horizontal resolution, laser range are also important and related to the error. laser scanner setup parameter such as pitch and roll are important as well.

  - **Down sampling of input scan**

    Generally, in the matching algorithm, input scan is down-sampled for two reasons. One is to decrease the computation complexity and the other is to make the input scan more even. Some of the features might be removed or might become more uncertain in this phase. Therefore, down-sampling can be one of the error source.

- **Distortion of the input scan**

  Usually this distortion is happen when the vehicle is moving. The error caused by this phenomena is related to the sensor which is used. If sensor is SICK then the frequency of each scan is more than 50 Hz. If the vehicle speed is 36Km then in each second it moves 10m. Thus in the time of emitting one single beam the vehicle only moves 20cm. So there is a 20cm shift between the beginning point of the scan and the endpoint. This 20cm has minor effect in localization. However if the scanner is Velodyne, then its frequency is usually 10Hz. In this case, if the vehicle moves with 36Km, in each scan the vehicle moves 1.0m. Therefore, the beginning point and endpoint of the scan has 1.0m shift. This means that the scan points has average shift of 50 cm and thus localization error might be near 50cm.

- **Matching algorithm**

  - **Matching algorithm robustness and accuracy**

    Matching algorithm can be one of the error source for self-localization. Some of the algorithm are more robust to local optimum and some of them are more robust to partial overlapping. Some of the algorithm can register two different shapes of scan and some cannot. As the density of map and input scan is not same matching algorithm should be more robust to this phenomena.

  - **Initial guess**

    For map-matching techniques before the matching of input scan to the map, initial guess of the input scan should be defined. Initial guess has a huge effect on the matching especially for the optimization based techniques because if the initial guess is not close enough, it might stuck on the local optimum. This initial guess is very important in the NDT-based method as well.

- **Dynamic phenomena**

  In some part of the map we might have environment change. For example, some building has newly built or some building destroyed. Some of the changes are seasonal. Usually trees cause these kind of huge environment change. If the map is generated in summer that trees has leaf and map is used in winter then there is a huge environment change and this cause localization error. Sometime this environment change happen by dynamic objects. In the mapping phase there are cars parked in the streets but in the localization phase, the previously parked car is no longer there or the position of the car is changed. This kind of small environment change does not affect the localization so much. However, presence of a bus or a truck in the mapping phase can cause blockage of laser scanner and hole in some part of the map.

- **Map**

In map-matching methods map plays a key role in localization accuracy. This role is explained in detail later in this chapter.

In the map based categories, map plays a significant role in achieving high accuracy self-localization. For accurate self-localization, the global and local accuracy of the map is essential as well, and many types of research have been done to obtain such a highly accurate map [3], [6]–[8]. However, the highly accurate map does not guarantee the accuracy of the localization [1]. In other words, map accuracy is different than the ability of the map for localization. For example, in the case of the tunnel, no matter how much the map is locally and globally accurate, the lack of longitudinal features in the map causes localization error in moving direction.

To achieve accurate self-localization within a map, the map should satisfy some requirements. In other words, the map should meet some specific criteria which define the ability of the map for self-localization. To the best of this author’s knowledge, there is no comprehensive study of the definition and formulation of these criteria. Therefore in this chapter, the required criteria regarding the ability of the map for accurate self-localization are defined and for each of the criteria, several map factors are defined.

Some of these criteria highly related to the environment. For example, in the city environment, there are more structured artifacts than rural places or crop fields. Thus the features for the map-matching can be found easier, and as a result, the localization become more accurate [9]. Tunnels, urban canyons, and highways can be assumed as another example in which environment are not suitable for self-localization using map-matching. In these scenarios, the vehicle is surrounded by two long walls in each side. Thus, the features needed for longitudinal positioning are not enough, and as a result, there will be an error in moving direction. Surrounding environments are different from place to place in the map and should be evaluated by defined criteria.

Some other criteria are related to the quality of representation of the environment by the map. In addition to the sensor related parameters in the mapping phase such as frequency of laser scanner, layer count, the range of the beam, setup parameters, etc., quality of representation highly related to the format and abstraction ratio (resolution) of the map. Many types of research have been done to propose an abstracted map format to both reduce the map size and the computational complexity [10]–[17]. Quality of representation of the environment in each of the map formats are different. In other words, some map formats discards more details of the map comparing to others. This information loss of the map might change the quality of some of the criteria and lead to localization error [18]. However, in some part of the map, abstraction does not necessarily change the quality of the map, or rate of change is acceptable. Additionally, in some cases, other criteria might compensate the lack in one criterion. Therefore, in order to evaluate the self-localization ability of the map in a specific point, all criteria should be considered together.

In this chapter, four general criteria for the map are defined. These criteria are feature sufficiency, layout, local similarity, and representation quality of the map. These criteria are defined regardless of the map format and can be applied to any other map formats. However, in this chapter, to quantify each of these criteria, the focus is made on the ND map format, and several factors are defined. For each point in the map, these factors are calculated based on the features in surroundings called local vicinity. Overview

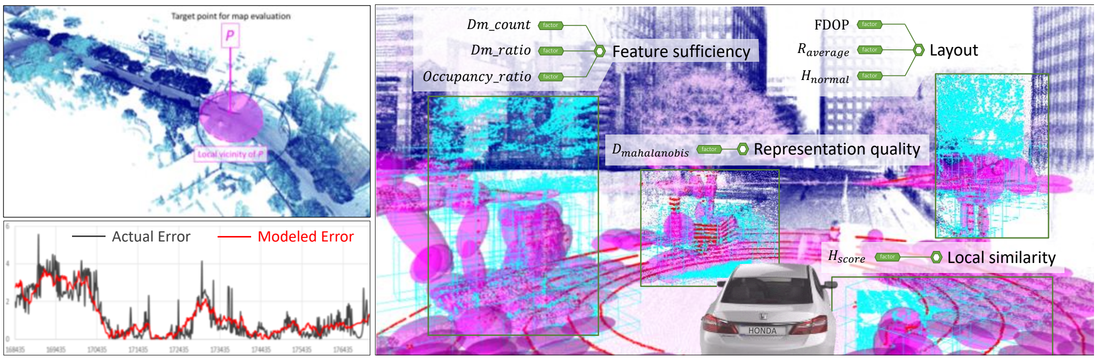


Figure 5-1 . Overview of the proposed map evaluation method. For a sample point P (top left), the map factors in the local vicinity are calculated (right). Based on these factors, the localization error is estimated (bottom left).

of the proposed map evaluation frameworks are shown in Figure 5-1.

As far as the knowledge of this author, there is very few works that is related to the map evaluation. Some of them are explained here. As already mentioned, one of the challenges of the optimization based map matching methods such as NDT is its sensitivity to the initial guess. This sensitivity is different from place to place in the map. In [6], to predict the localization error in different place of the map from different initial guesses, experimentally determined uncertainty is used. In their method, a vehicle once traverses whole paths, and an error model is calculated empirically. Later in the localization phase, this error model is used for rectification. One of the drawbacks of this method is that it needs a high amount of field experiments. Contrary to this method, using proposed factors in this work, localization error can be modeled without a field test. However, error model in [6] directly consider the input scan quality such as density of the scan as well, while in this work, input scan quality is inferred from map factors and for sure former can model more precisely.

In [73], the author used the building information from the open street map (OSM) as background knowledge to predict the erroneous part of the paths to improve the graph-based SLAM. In this work, point cloud map are considered to be available. Thus, the factors are directly extracted from the point cloud. OSM does not contain detailed shape of the building or irregularity in the building walls which highly affect the map matching results. In [74] author calculates the quality of map for visual odometry by heuristically estimating the entropy of the map. In [75] the author used the 3D polygons for self-localization. They use the correlation of the polygons to detect the localization accuracy beforehand.

In the field of scan alignment detection, some works have been done which is worth to mention and compare with our method. The reality is that, no matching algorithm exist today that can provide any certain methods for detecting misaligned point clouds [76]. And as experimental results in [76] shows, NDT and ICP both will not fail in the same point cloud set [15]. In the much less structured environment number of failed registrations are higher [15]. Therefore, in [76], H. Almqvist and M. Magnusson compared several geometric consistency methods for classifying aligned and nonaligned point cloud pairs. They also combined these classifier together with Adaboost to make stronger classifier. The evaluations shows the combined measure achieve better results than the single measures at the cost of requiring a large training set [15]. Some of the measures used in matching is : RMS(which is used in ICP), NDT score (which is used in NDT), NDT Hessain, and Plane Extraction, Partitioned mean normal [77], and surface interpenetration measures [78]. Alignment quality of point clouds can be investigated at different levels, from maps consisting of many point clouds to point cloud pairs [15]. Evaluation are done on four measures and a combined measure was performed using multiclass support vector machine [79].

But there are some key difference between scan to scan alignment evaluation and map evaluation. These differences are as follows:

- One of the difference of scan to scan matching misalignment detection and map evaluation using factors is that in the former, they use some measure between those point clouds. It means the scan is already exist. However, in our method, we assume that we only have the map, and only using map data, map is evaluated.

- We focused on the environment rather than the scan itself. We think that this is the environment that cause this misalignment.

- One more difference between these two is, the map and the scan shape is totally different. Map has denser data and scan has very sparse data. Thus if we want to use some of the measure mentioned in [15], we cannot perform the evaluation. Data set used in both [15] and [79] consist of scan made by single layer laser that upward and rotates, thus the density of the scan is high.

Outcomes of this chapter is evaluated in chapter 6 and applied to the adaptive determination of map and localization parameters in chapter 7.

One of the important outcome of this research is that, it can be applied to evaluate the capability of the map provided by mapping companies for the self-driving localization application and fills the gap between mapping companies and automobile industries. In fact, some of the digital maps are already made for the other purposes. For example Japan government already has a GIS map which is a vector map. The purpose of this 2D map is not autonomous driving. Therefore, in order to use it as a self-localization purpose, the GIS map can be evaluated by applying the map evaluation factors.

The other benefit of this map evaluation method is that, those part of the map which cannot meet the thresholds are detected. After detection, there are two possible solution. One is to tell the government to put some artificial objects there to increase the map evaluation factors or to tell the autonomous vehicle companies to use other sensors for doing self-localization in that region of the city. Indeed, this map evaluation can be also performed in an experimental way or field test. In this way, after the map is generated, the car drive the part of the map that needed to be evaluated and compared the self-localization output to the ground truth. By comparing the self-localization output with the ground truth we can evaluate the accuracy of self-localization and ability of the map for self-localization. However this method has two problems. First, if we want to perform the field test for all drivable area in the map, it needs so much time and cost for field test evaluation. Second, after detecting the erroneous part of the map, we cannot determine which factor cause this error to solve it.

For the map matching based methods, in addition to the map related factors, other parameters which is not related to the map can also affect the self-localization accuracy which is ignored in this work.

The rest of this chapter is organized as follows. Section 5-2 describes the map evaluation criteria, and section 5-3 formulates each of the criteria using map factors.

5.2. Evaluation criteria

In this section criteria for a map evaluation are defined. These criteria are feature efficiency, layout, local similarity, and representation quality of the map and defined regardless of map formats. Overview of the criteria is shown in Figure 5-2.

5.2.1. Feature sufficiency of the map

In self-localization techniques based on map-matching, pre-built map is made up of features. The type of features varies depending on the map formats and matching algorithms. For example, in the point cloud map the features are points, and in the occupancy map, features are the occupied cells. Likewise, in the ND map the features are NDs, and in the case of vector map and planar surface map, the features are vectors and planar surfaces respectively. For better self-localization accuracy, plenty yet high-quality

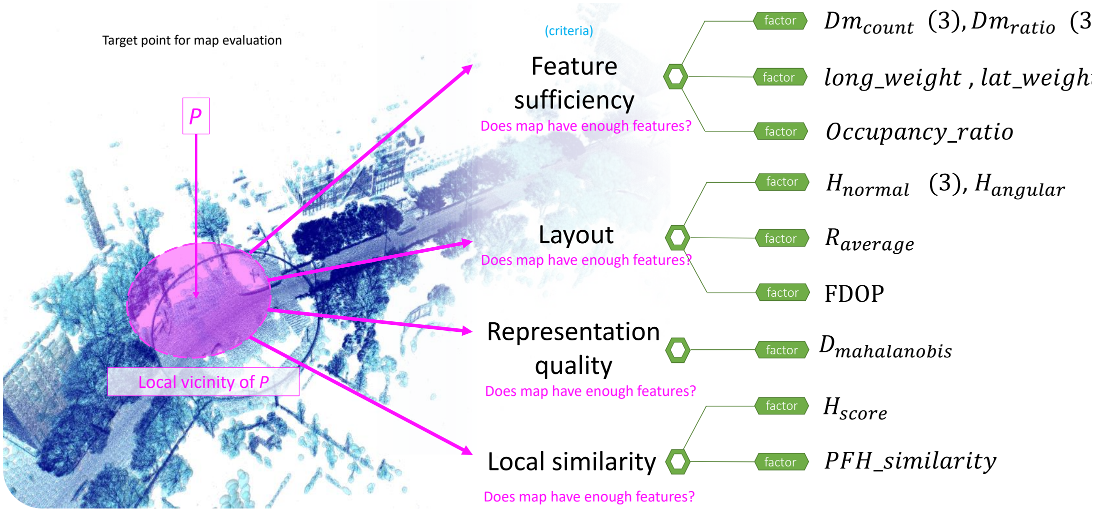


Figure 5-2 . Different layout for the map. In (A) all features are lean to one corner. This layout can cause positioning error. In (B), the distribution of the features are even.

features are required. The quality and number of extracted features are related to the environment. For example, in the urban city, there are many buildings and structured environments that produce plentiful features in the map. However, in the crop fields, there are no sufficient features for matching. Therefore generally in the urban area, the accuracy of localization using laser scanner is higher. In the case of a planar surface map, in the environment with many vegetation and trees, extraction of planar surfaces is difficult, and thus the features for matching are not sufficient.

In addition to the number of features, the quality of the features is important as well. In some part of the map, there might be plenty of features; however, as the quality of them are not adequate, the localization cannot be performed accurately. Quality of the features varies in different map format as well. Consider ND map as an example. In order to make ND map, the space of the map are subdivided into static voxels. Then, each voxel is represented by the normal distribution calculated from the position of the points. In the matching phase, these normal distributions lead the input scan point cloud to its optimum location. In ND map, the shape of the normal distribution can be assumed as the quality of the features, and it might play an important role in optimization.

To formulate the feature sufficiency, three factors are proposed. These factors are $feature\_count$, Dm_ratio, and $occupancy\_ratio$. These factors are described in section IV.

5.2.2. Layout of the map

In addition to the count and quality of the features, the layout or displacement of the features in the space might be important. In some part of the map, there might be plenty of high-quality features. However, as the features are all placed in one corner of the map, the quality of matching degrades (Figure 5-3(A)). In Figure 5-3, blue circle in the center shows the vehicle position and small red circles show the features of the map which are captured by a laser scanner. In Figure 5-3(A), captured features in the map are distributed evenly. However, in Figure 5-3(A) the features of the map are placed in the upper left corner of the vehicle. This uneven distribution of the map features can cause the localization error.

In fact, the concept of the layout of the features comes from a global positioning system (GPS). In the GPS-based localization, if satellites are not distributed evenly, in other words, satellites lean in one region, then the calculated position will be erroneous. This is because each of the pseudo-ranges calculated from each satellite has an error by itself. If all satellites are lean in one small region, then they cannot compensate each of the pseudo range error, and the input error highly appear in the output. For GPS, the ratio that shows how much the input error affect the output localization result is formulated by geometrical dilution of precision (GDOP)[80]. We inspired from GDOP and defined a map factor called feature DOP ($FDOP$) to formulate the distribution of the features in the map space. GDOP and satellite position is shown in **Figure 5-4**(b). In scenario 1 the distribution of satellites are even so the GDOP is very small. It means that the input error

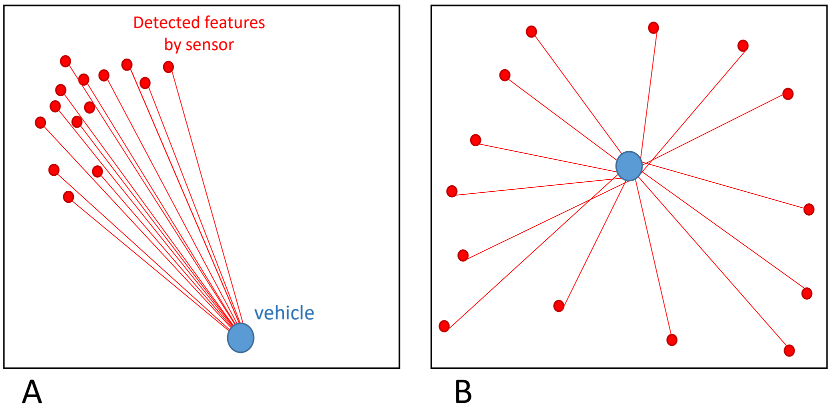


Figure 5-3 . Different layout for the map. In (A) all features are lean to one corner. This layout can cause positioning error. In (B), the distribution of the features are even.

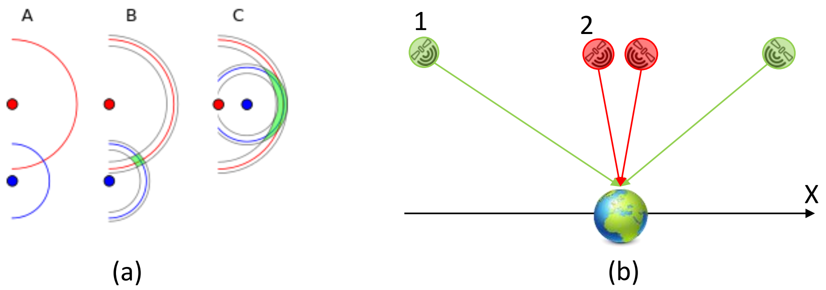


Figure 5-4 Effects of satellite distribution to the GDOP.

will not result in high output error. However in scenario 2, as the satellites are lean more closely the GDOP is high. This means that the small error in pseudo ranges results in high positioning error.

In addition to the distribution of the features, in some part of the map, there might be a situation such as tunnels, urban canyons, and highways in which the layout of the buildings is not suitable for lo ngitudinal positioning (Figure 5-6). In these situations, the layout of the buildings is so that the lateral position of the vehicle can be obtained accurately but not longitudinal. As shown in Figure 5-5(A), when the vehicle is in the position A, the layout of the buildings is so that it cannot get the features for longitudinal positioning. In position B, the situation is same. However, in Position C, the layout of the buildings changed and the vehicle is able to observe the features for longitudinal positioning. In both A and B position, the previously defined criteria (feature sufficiency) are satisfied as the car is surrounded by plentiful high-quality features. We

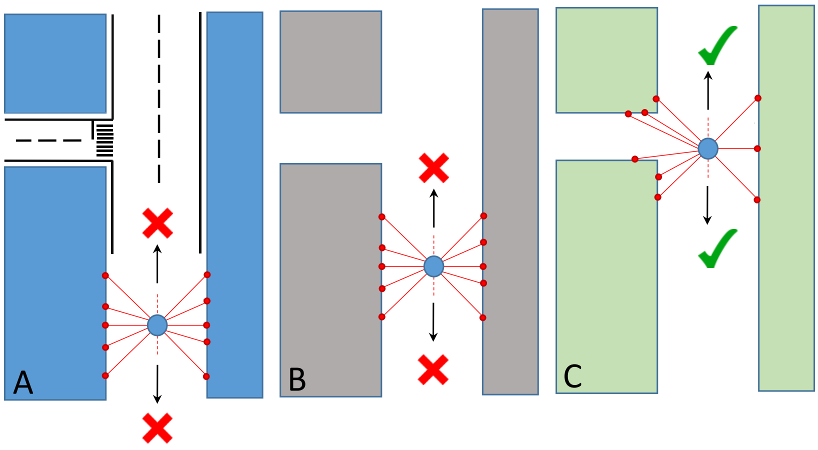


Figure 5-5 Different situations (A and B) in that the localization in longitudinal direction has error.

introduce $normal\_entropy$ factor to formulate such kind of layout problem in the map.

To formulate the Layout criterion of the map, in addition to aforementioned factors, $r\_average$ and $angular\_entropy$ are proposed and described in section IV.

5.2.3. Local similarity of the map

There are some positions in the map that the environment has many similar features (Figure 5-6). These similar features make the positioning difficult. In Figure 5-7, blue shapes are a map, and red lines show the observation of the laser scanner. In this case, as there are similar shapes in the map, it is impossible to detect which is the actual vehicle position; A or B. In other words, in both position A and B the observation of the laser scanner are same. In this situation, the map has local similarity, and this criterion is not satisfied. Again, in this scenario, there are plenty of features, so the feature sufficiency is met, the distribution of the features and layout of the buildings are satisfactory, so the layout criterion is satisfied as well. However, due to the presence of local similarity in this part of the map, localization is erroneous.

Environment and map abstraction both can cause local similarity. If the abstraction ratio become high, then some details of the map are eliminated, and consequently local similarity increases.

In order to formulate the local similarity, $score\_entropy$ is proposed. Formulation of $score\_entropy$ is described in section IV.

5.2.4. Representation quality of the map

One of the important criteria for the map which is directly related to the map format and abstraction ratio is the representation quality of the map. Representation quality

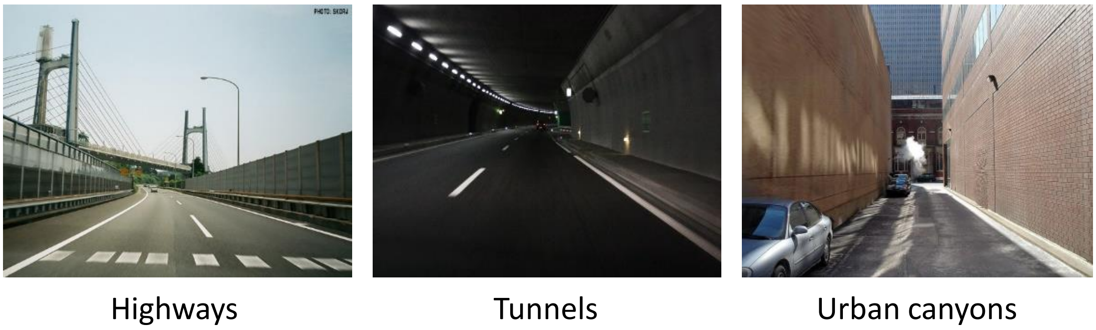


Figure 5-6 Situation that the map does not have any lateral feature to get the longitudinal position.

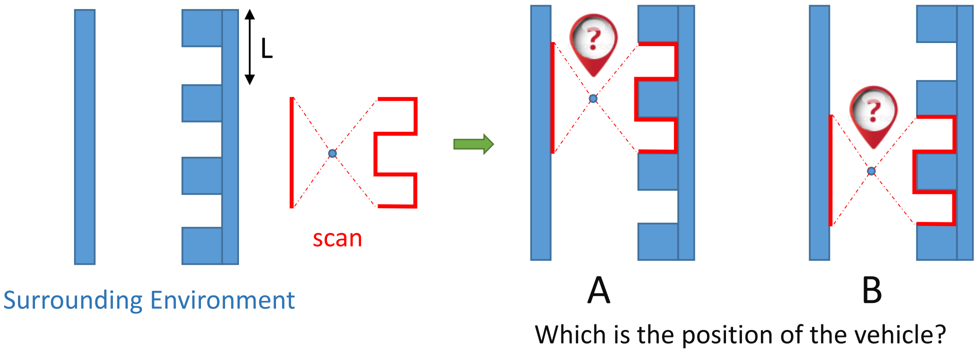


Figure 5-7 Local similarity cause the positioning uncertainty. Longitudinal position cannot be obtained. The blue shapes are the buildings which surround the laser scanner. The red lines are scan on the specific time and shows the observed part of the building by laser scanner. In this scenario, vehicle cannot distinguish between position A and B.

shows how much the generated map is similar to the actual environment.

On the one hand, representation quality of the map is related to the type of laser scanner and sensor parameters in the mapping phase. These parameters are number of layers, the rage of beams, scanning frequency, number of sensors, and setup parameters such as pitch and roll angles. As an instance, raw point cloud collected by Velodyne HDL-64 which has 64 layers of laser beams can represent the environment more densely comparing to the VLP-16 which only has 16 layers. However, in this work, effects of mapping phase to the quality of representation are ignored.

On the other hand, representation quality related to the map format and resolution of the map. In fact, the representation quality criterion shows how the final map is similar to the raw point cloud map collected in the mapping phase. Low representation quality of the environment might change the peak of score function. If the peak is changed, correct answer from optimization process will be the wrong answer for the localization. In Figure 5-8 change of peak after abstraction in ND map format is shown. Figure 5-8(top) shows the score of matching for 2.0m grids and below shows the matching for the same region with 4.0m grid size. Score peak should correspond to ground truth. However, in 4.0m grid size, as more abstraction applied to the map, the peak of scores are shifted 60cm. Consequently, even if the optimization process does not stick in local maximum, localization has a 60cm error.

Abstraction does not always change the score peak. Our investigation shows for

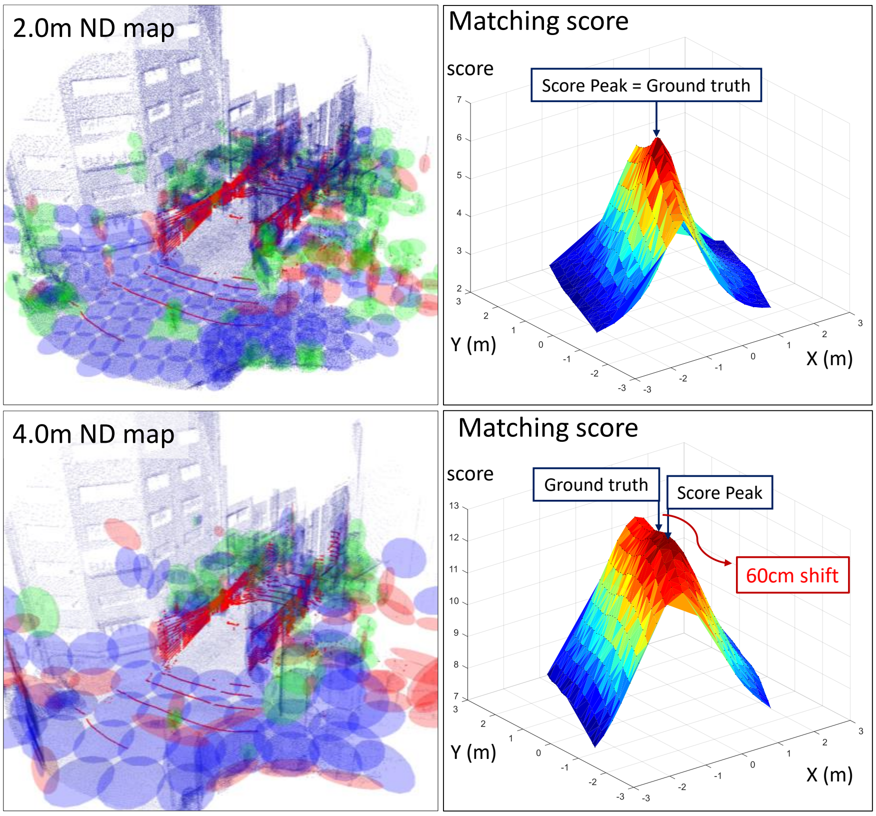


Figure 5-8 Shift of score peak due to abstraction of the map. Top images shows the 2.0m ND map and the matching score of scan (red points) on it. Score of 2.0m ND map does not have any shift. Bottom image shows the same pair for 4.0m ND map. In 4.0m ND map, due to higher abstraction of the map, peak of matching score is shifted by 60cm. This shift of peak at least cause 60cm localization error.

2.0m grids, about 77.3% and for 4.0m grid about 43.1% of the cases, peak remains unchanged. This is because abstraction always does not affect the representation quality. Consider a ground or a flat wall. No matter how much the size of grid expands, the representation quality remains same. Therefore, to evaluate the mapping capability for localization, there should be a factor to evaluate the map representation quality. $mahalanobis\_distance$ factor can formulate the criterion for representation quality and explained in section IV.

5.2.5. Formulation of the evaluation criteria

In this section, to quantify each of the proposed criteria for the map, several factors are defined. Map evaluation criteria are defined regardless of map format. However, to formulate each factor, self-localization based on normal distribution transform (NDT) which is well-known in the field of vehicle self-localization is considered.

In NDT methods, instead of directly registering the Velodyne scan to the point cloud map, the scan is registered to the set of normal distributions called ND_{map} which is made from point cloud map. In order to make ND_{map}, map space is subdivided into fixed size voxels called cells. Then, from the points inside each cell, normal distribution (ND) is calculated. To register the scan on this NDs, the scan is moved over the NDs with 6D transformation matrix T and for each pose, likelihood are calculated. The scan pose correspond to the maximum likelihood is obtained by Newton's optimization algorithm and considered as the position of the vehicle within a map. For further discussion readers are referred to [11], [70].

To calculate the factors for position P, first, the map elements (normal distributions) in the range of laser scanner beam from P are extracted from the map. Second, the elevation angle ψ of each map elements to point P is calculated. In this work, for the localization, Velodyne VLP-16 is used. The elevation angle range of VLP-16 is -15 to +15 degree. Therefore, from the map elements extracted beforehand, those are considered with the elevation angle $-15 < \psi < +15$. These elements are called local vicinity of the position P in the rest of this work and shown in Figure 5-9. Factors for position P is calculated based on the local vicinity of it.

Point cloud correspond to local vicinity is represented by $S = \{x_1, x_2 \dots x_n\}$, where n is the number of points. NDs correspond to local vicinity is represented by $ND_{map} = \{N(\mu_1, \sum_1),\ N(\mu_2, \sum_2) \dots N(\mu_m, \sum_m)\}$, where m is the number of NDs, and μ and $\sum$ are the mean and covariance matrix of the normal distributions respectively. Figure 5-9(b) shows the raw point cloud of the extracted region. Figure 5-9(a) shows the ND_{map} of the same region which is used for calculation of the factors.

5.3. Factors for feature sufficiency

The first criterion is feature sufficiency of the map. In order to formulate feature sufficiency, three factors are proposed. These factors are Dm_count, Dm_ratio, and $occupancy\_ratio$.

5.3.1. $feature\_count$, Dm_count, and Dm_ratio

The first factor is $feature\_count$. $feature\_count$ shows the number of map features in the local vicinity. In the case of ND map format, features are normal distributions. More feature count should lead to less localization error.

In order to evaluate the quality of the features in ND_{map}, the dimension value of the normal distributions are considered. The dimension value of each ND are calculated inspired by [81] as follows:

First, for each normal distribution, Eigenvalues and Eigenvectors are calculated. Consider that the Eigenvalues are λ_1, λ_2, and λ_3.

$$\textstyle\sum_j = \mathrm{U}\Lambda U^T, \tag{1}$$

Where $\sum_j$ is covariance matrix of the Normal distribution j and U is Eigenvectors and Λ is Eigenvalues as follows:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}, \tag{2}$$

From Eigen values the standard deviations are calculated as follows:

$$\forall i \in [1,3] \;\; \sigma_i = \sqrt{\lambda_i}, \tag{3}$$

where i is the index of Eigen values. Using standard deviation, three dimension behavior are defined as follows (σ_i is sorted so that $\sigma_1 > \sigma_2 > \sigma_3$):

$$a_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1}, a_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1}, a_{3D} = \frac{\sigma_3}{\sigma_1}, \tag{4}$$

where a_{nD} is the n_{th} dimension behavior of the feature. If $a_{1D} \gg a_{2D}, a_{3D}$, the

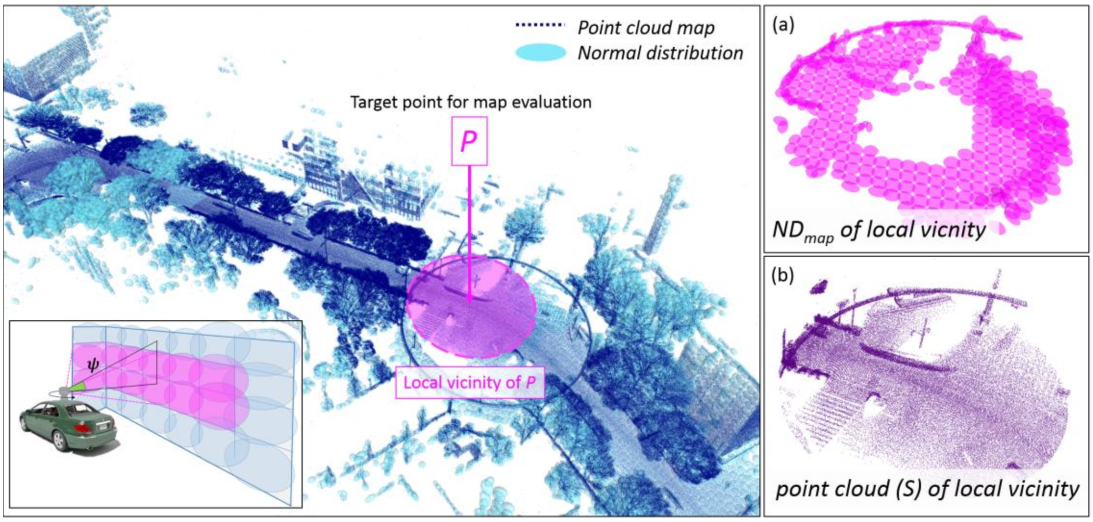


Figure 5-9 Local vicinity of target point P. Local vicinity is extracted based on the range and vertical angle ψ of the laser scanner. (a) ND map of local vicinity. (b) point cloud map is the local vicinity.

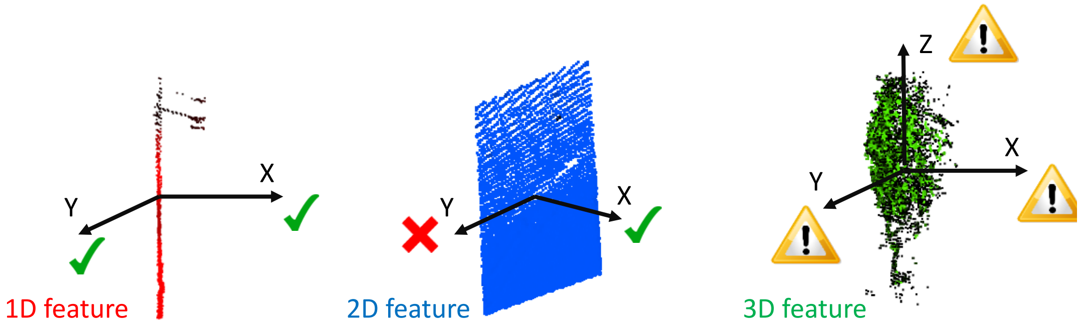


Figure 5-10 Definition of 1D, 2D and 3D features. 1D feature can perform the positioning for 2 dimension. 2D features can perform in 1 dimension and 3D object has high uncertainty so it cannot get the correct location in any direction.

feature considered to be $1D$ feature which is pole-like features (Figure 5-10 (left)) and if $a_{2D} \gg a_{1D}, a_{3D}$, the feature is $2D$ feature which is more like walls and planar surfaces (Figure 5-10 (middle)). Finally, if $a_{3D} \gg a_{1D}, a_{2D}$, then the feature is $3D$ feature which is more like vegetation or scatters (Figure 5-10 (right)).

As already described in the Chapter 2, the normal distribution is used to lead the input scan to its optimum position. The contribution of each ND to this lead is defined by their shapes. If the shape is more like sphere (3D feature), it means that ND is more uncertain and the portion of contribution is less. In other words, it cannot help so much for adjusting the input scan to its proper position. Having 3D features increase the uncertainty of the whole map, however having of it, is better than having no features. 2D feature are plane-like features and can lead the input in one dimension. It can lead the scan to the direction parallel to its normal. As shown in Figure 5-10(b) it can only correct the position of the wall in the X direction. In other words, the uncertainty of 2D-ND features are high in 2 dimension but in one dimension which is X dimension, the uncertainty is very small. Thus it can be used to align the scan in that dimension. These features (2D-NDs) are more useful in the matching and thus in the localization comparing

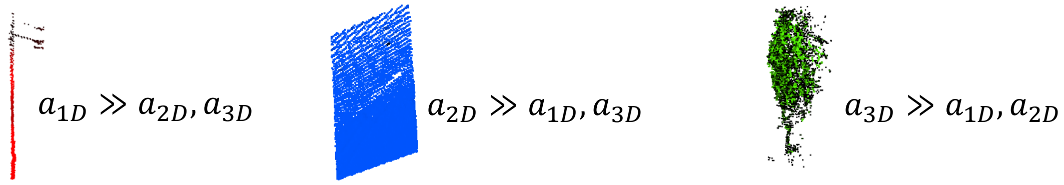

$a_{1D} \gg a_{2D}, a_{3D}$
$a_{2D} \gg a_{1D}, a_{3D}$
$a_{3D} \gg a_{1D}, a_{2D}$

Figure 5-11 feature dimension value for three types of objects.

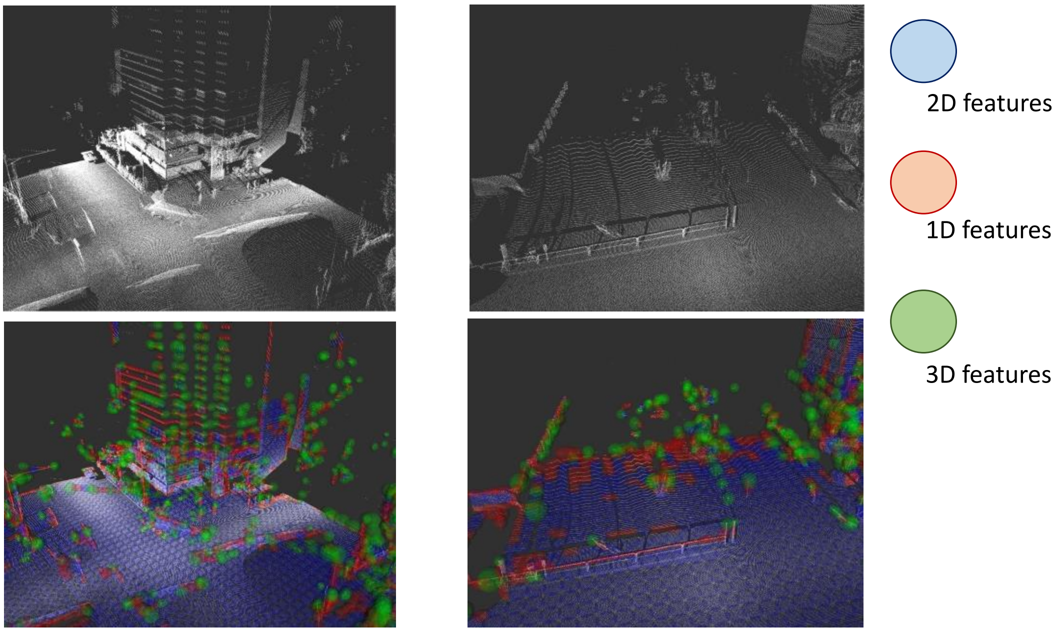


Figure 5-12 1D, 2D and 3D shapes in point cloud map. The red features are 1D features and the blue features are 2D features and the green s 3D features.

to 3D feature. And finally 1D features has the uncertainty only in one dimension. In other words, it has certain value in 2 dimension and can lead the input scan in 2 dimension to fit the map. 1D features is shown in Figure 5-10(a). As shown in the figure, this type of feature can lead the input scan in X and Y dimension. It cannot fit the position of input scan in Z dimension. 1D features are more useful than both 2D features and 3D features. Usually vegetation and trees and specially scattered objects makes 3D features. Building walls can make 2D features and poles and edges make 1D features.

Figure 5-13 shows different dimension property of the features with different colors in the map. In Figure 5-13, red features are $1D$, blue is $2D$ features, and green features are $3D$ features. As an instance ground and building walls form $2D$ features and lighting poles and tree trunks are forming $1D$ features which is red. However, sometimes due to the sensor parameters and sensor setup in mapping phase, the walls point cloud appear sparse and thus more 1D features appear in the map. This is shown in the Figure 5-12. Using this definition and the property of the features, $feature\_count$ factor are divided into $D1\_cnt$, $D2\_cnt$, and $D3\_cnt$ factor. Dm_cnt factor ($\forall m \in [1,3]$), shows the number of Dm features in the space. As an instance, $D1\_cnt$ shows the number of $1D$ features in the space. In addition to the number of whole features in the patch which can represent the sufficiency of the features, these factors can represent

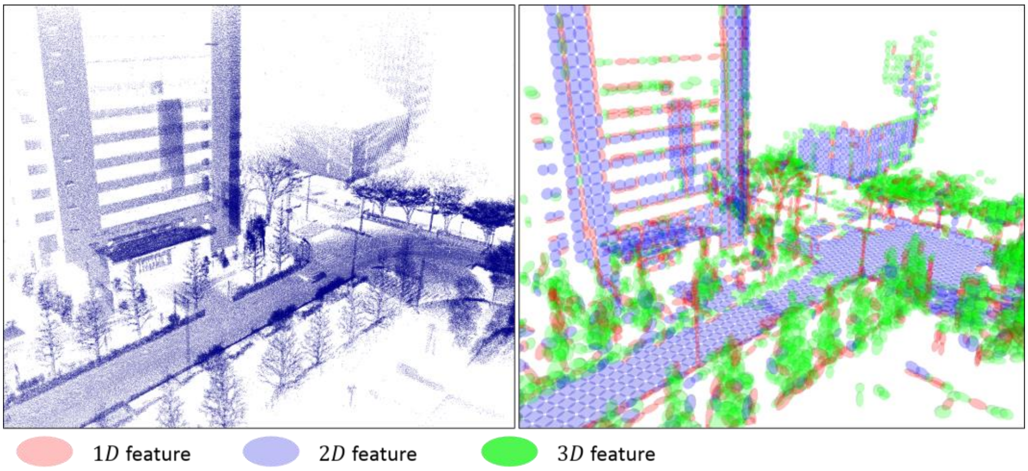


Figure 5-13 1D (red), 2D(blue) and 3D (green) features in map. Left image is point cloud amp and right image is the ND map generated from left point cloud with 2.0m grid. Ground and building walls are 2D features and poles and trunk of trees are 1D and vegetation and trees has 3D features.

the quality of the features in the patch. In addition to the Dm_cnt factor ($\forall m \in [1,3]$), the ratio of each feature over all features are considered as another factor. These factors are $D1\_ratio$, $D2\_ratio$, and $D3\_ratio$.

$$\forall i \in [1,3],\ Di\_ratio = \frac{Di\_count}{D1\_count+D2\_count+D3\_count}\ , \tag{5}$$

5.3.2. $occupoancy\_ratio$

The next factor which can formulate the feature sufficiency criteria is $occupancy\_ratio$. $occupancy\_ratio$ shows how much the surrounding environment which can be seen by laser scanner is occupied with the map features. This factor consider the feature sufficiency from the view of LiDAR instead of 3D space. In previously defined factors such as $feature\_count$, some of the features counted on the factors might be redundant because they are not observed by the laser scanner. Laser scanners can only capture the closest object in the scene.

In order to consider this characteristic of the laser scanner, the local vicinity space is converted to the local vicinity depth image as shown in the bottom of Figure 5-14 (b).

This depth image of the map can be seen as 2D histogram too. Vertical and horizontal resolution of the depth image is set to the number of layers and horizontal resolution of the laser scanner respectively. In this work, for localization, Velodyne VLP-16 is used. This laser scanner has 16 vertical layers with resolution of 2°. Therefore the

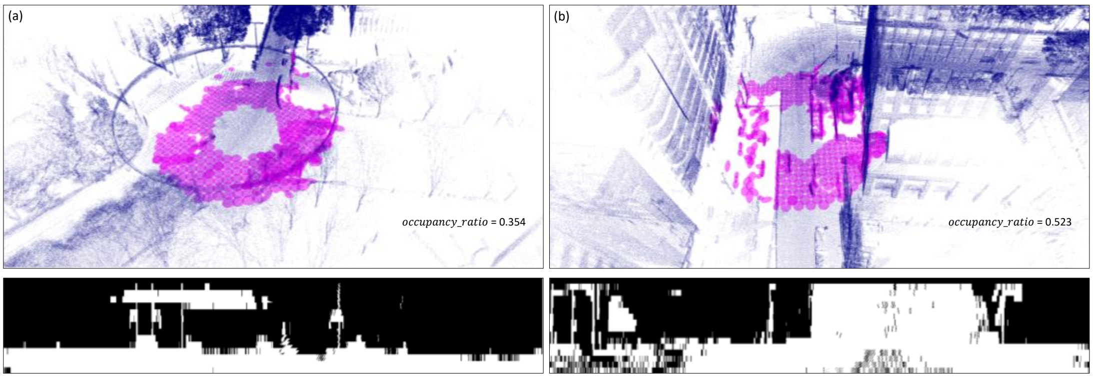


Figure 5-14 Depth images (bottom images) correspond to the features in local vicinity (purple ellipsoids). In depth image white shows presence of features and black shows the absence of features. Occupancy ratio for the left image is 0.354 and for the left is 0.523. In right scenario, local vicinity is occupied with more features thus the occupancy ratio is higher.

vertical resolution of the depth image is set to 2. Rotation frequency of the laser scanner directly related to its horizontal resolution. The rotation frequency of VLP-16 set to 20Hz (1200 rpm). Horizontal resolution of the VLP-16 in this rotation speed is 0.4°. Therefore the horizontal resolution of the depth image is set to 0.4 as well.

For each normal distribution in the local vicinity, the vertical angle Ψ and horizontal angle (azimuth) Φ to point P are calculated. According to these two values, the depth image are filled with 0's and 1's. 0's in the depth image mean that there is no corresponding feature in that specified angel and 1's shows the presence of the feature. In Figure 5-14 (b) black shows the absence and white shows the presence of the features.

$occupancy\_ratio$ factor is the ratio of the occupied cell over all cells of the depth image and calculated as follows:

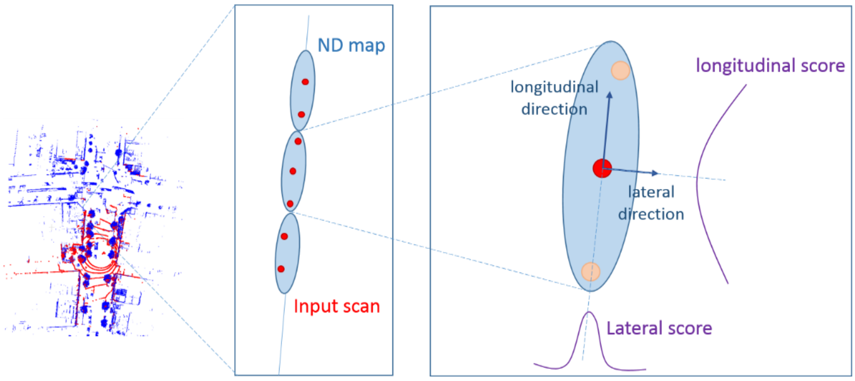


Figure 5-15 Quality of localization in longitudinal and lateral is depend on the direction of feature's normal. Lateral score and longitudinal score is shown with purple. Uncertainty in lateral is lower, thus lateral weight should be higher than longitudinal weight. Blue ellipsoid shows the ND_map and red points is input scan.

$$occupancy_{ratio} = \frac{occupied\_cell}{all\_cells} \quad , \tag{6}$$

5.3.3. $long\_weight$ and lat_weight

In order to evaluate the quality of the feature for self-localization, $long\_weigh$ and lat_weight are defined. $long\_weigh$ measures the quality of the feature for longitudinal positioning and lat_weigh measure the quality of feature for lateral positioning. In Figure 5-15 ND features and one point from scan (red point) is shown. In this scenario, normal of the ND feature is perpendicular to the longitudinal direction. Uncertainty of the feature in longitudinal direction is high. Thus in this direction the localization accuracy is low. On the other hand, uncertainty of the feature in lateral direction is low. Thus the score function in lateral direction has sharper peak and thus the accuracy in lateral direction is higher. In this scenario, lat_weigh is high and $long\_weight$ is high. Longitudinal and lateral weight of each feature x_i are calculated based on the degree of the angle between normal of the features and longitudinal direction $\emptyset_i$, and the distance of the feature to the center of local vicinity r_i. The importance of r is described in the section of layout criterion. The $long\_weight$ and $long\_weight$ factors are calculated as follows:

$$long\_weight \ = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i}sin(\emptyset_i) \quad , \tag{7}$$

$$lat\_weight \ = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i}cos(\emptyset_i) \quad , \tag{8}$$

Where n is the number of features in local vicinity.

Longitudinal and lateral direction is not important. Any direction can be assumed as longitudinal and the perpendicular direction to longitudinal is lateral direction. In this work, longitudinal direction is assumed to be same as road direction. Here several factors are calculated from this weight. $weight$ factor is the summation of all weights and calculated as follows:

$$W_{lon} \ = \sum_{i=1}^{n}\frac{1}{r_i}sin(\emptyset_i), \quad , \tag{9}$$

$$W_{lat} \ = \sum_{i=1}^{n}\frac{1}{r_i}cos(\emptyset_i), \quad , \tag{10}$$

$$weight \ = \sqrt{{W_{lon}}^2 + \ {W_{lat}}^2} \quad , \tag{11}$$

$$weight\_avr \ = \frac{1}{n}\sqrt{{W_{lon}}^2 + \ {W_{lat}}^2} \quad , \tag{12}$$

Where n is the number of ND features in local vicinity.

contribution of each feature to leading the input scan to its correct position is related to the shape of them. Features with smaller Eigenvalues has smaller uncertainty and can lead to better registration solution. Thus, in addition to the aforementioned factors, $long\_weight\_eigen$ and lat_weight_eigen is defined. Corresponding Eigenvalues for the normal are assumed to be the weight of this feature. If the Eigenvalues are big, it means the uncertainty is high, and if the Eigenvalues are small, it means the uncertainty are low and the weight should be high. The $long\_weight\_eigen$ and lat_weight_eigen are calculated as follows:

$$long\_weight \ = \frac{1}{n}\sum_{i=1}^{n} K \frac{1}{\lambda_i r_i} sin(\emptyset_i) \quad , \tag{13}$$

$$lat\_weight \ = \frac{1}{n}\sum_{i=1}^{n} K \frac{1}{\lambda_i r_i} cos(\emptyset_i) \quad , \tag{14}$$

Where λ_i is the Eigenvalue correspond to i_{th} feature and K is a constant that make the range of λ_i and r_i almost same.

5.4. Factors for the layout

In order to formulate the layout criteria of the map, $FDOP$, $normal\_entropy$, and $angle\_entropy$ are defined.

5.4.1. $\boldsymbol{Feature\ DOP}$

Inspiring from GDOP in global positioning systems, Feature DOP ($FDOP$) is calculated. Assume that the center of local vicinity is $P = (x_p, y_p, z_p)$ and the position of the i_{th} ND feature is $f_i = (x_i, y_i, z_i)$ and its distance to P is r_i. Matrix A is obtained as follows:

$$A = \begin{bmatrix} \frac{(x_1-x)}{r_1} & \frac{(y_1-y)}{r_1} & \frac{(z_1-z)}{r_1} \\ \frac{(x_2-x)}{r_2} & \frac{(y_2-y)}{r_2} & \frac{(z_2-z)}{r_2} \\ \vdots & \vdots & \vdots \\ \frac{(x_m-x)}{r_m} & \frac{(y_m-y)}{r_m} & \frac{(z_m-z)}{r_m} \end{bmatrix} , \tag{15}$$

From multiplication of A and its transpose,

$$Q = A^T A = \begin{bmatrix} \tilde{\sigma}_x{}^2 & \tilde{\sigma}_{xy} & \tilde{\sigma}_{xz} \\ \tilde{\sigma}_{xy} & \tilde{\sigma}_y{}^2 & \tilde{\sigma}_{yz} \\ \tilde{\sigma}_{xz} & \tilde{\sigma}_{yz} & \tilde{\sigma}_z{}^2 \end{bmatrix} , \tag{16}$$

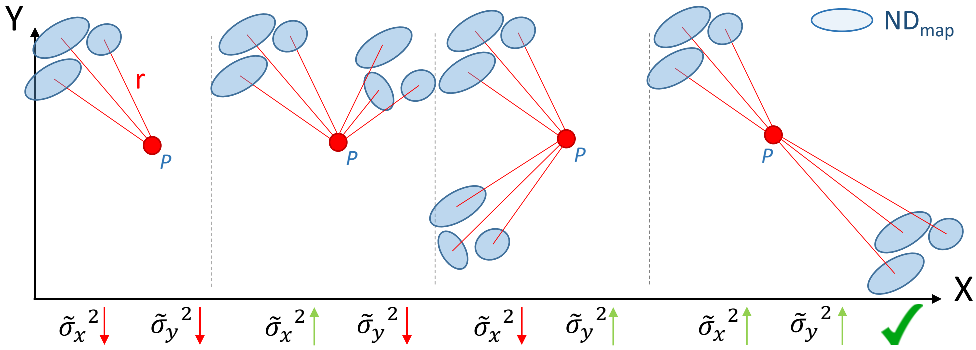


Figure 5-17 Comparison of the layout (distribution) of the feature with lateral and longitudinal FDOP.

$$A = \begin{bmatrix} \frac{(x_1-x)}{R_1} & \frac{(y_1-y)}{R_1} & \frac{(z_1-z)}{R_1} \\ \frac{(x_2-x)}{R_2} & \frac{(y_2-y)}{R_2} & \frac{(z_2-z)}{R_2} \\ \vdots & \vdots & \vdots \\ \frac{(x_m-x)}{R_m} & \frac{(y_m-y)}{R_m} & \frac{(z_m-z)}{R_m} \end{bmatrix}$$

Figure 5-16 The distance to features for different map format.

Q is obtained which can be assumed as covariance matrix of the position of the features related to the vehicle. Finally, $FDOP$ is calculated as follow:

$$QFDOP = 1/\sqrt{\tilde{\sigma}_x{}^2 + \tilde{\sigma}_y{}^2 + \tilde{\sigma}_z{}^2}\ , \tag{17}$$

If features are distributed uniformly in the space, the values of $\tilde{\sigma}_x{}^2$, $\tilde{\sigma}_y{}^2$, and $\tilde{\sigma}_z{}^2$ are high and thus $FDOP$ is low. $FDOP$ has a direct relation to the localization error. In Figure 5-16 The distance to features for different map format.

In Figure 5-17 several scenarios for FDOP are shown. FDOP related to the environment and abstraction ratio does not affect it.

5.4.2. *normal_entropy*

One of the important questions that should be answered in order to evaluate the layout criterion of the map is that, is the layout of the environment capable of providing features for both longitudinal and lateral positioning. Generally, in the matching based methods, the uncertainty of the positioning is related to the features normal.

In Figure 5-18 three scenarios with features with different normal are shown. Red arrows show normal of the features, and red ellipsoid around the car shows the positioning uncertainty. In Figure 5-18(a), all of the features are parallel to the moving direction of the vehicles. In this scenario, the longitudinal position cannot be calculated accurately, and the uncertainty in the moving direction is high. In Figure 5-18(b) the features are perpendicular to the moving direction. In this scenario, the longitudinal position can be obtained. However, the lateral position has a huge error. And finally in Figure 5-18(c), due to the presence of features in both direction, both longitudinal and lateral positioning can be obtained.

Consequently, if the features in the local vicinity face more directions, the positioning uncertainty decreases. $normal\_entropy$ factor can show the degree of dispersity of the direction of the features.

To calculate $normal\_entropy$, first, Eigenvalues and Eigen vectors of the NDs in the local vicinity are calculated. Second, for each ND, from Eigen values, normal is calculated. Corresponding Eigen vector to the smallest Eigen value are considered as the normal. Then the azimuth angle ϕ and elevation angle Ψ of the normal are calculated. According to ϕ and Ψ, the normal are stored in mxm bins histogram. In this work, the $normal\_entropy$ result are evaluated with $m = 8x8, 16x16$, and $90x90$ bins.

Finally, from this normal angle histogram, $normal\_entropy$ are calculated as follows:

$$H_{normal} = -\sum_{i=1}^{b}\sum_{i=1}^{b} P_{normal}\,(i,j)(log_2 P_{normal}(i,j)), \qquad (18)$$

where b is the number of bins for each axis, and $P_{normal}(i,j)$ is the probability of

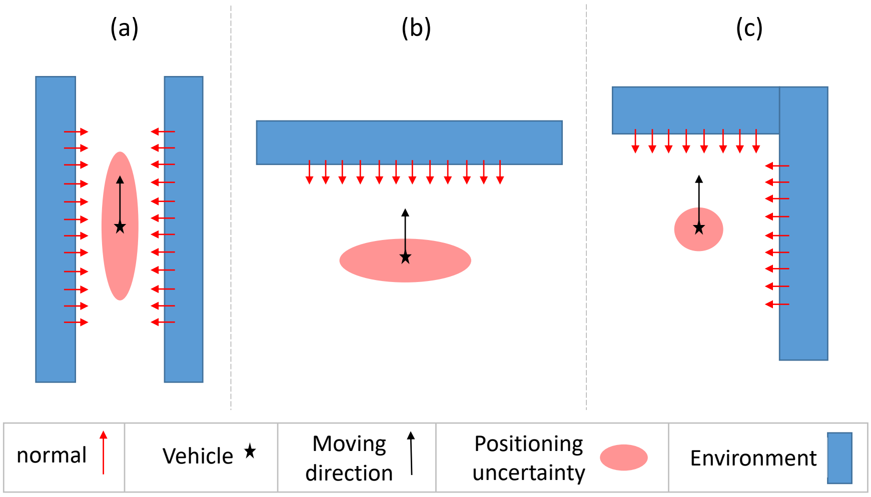


Figure 5-18 Vehicle position uncertainty (red ellipsoid) in different layout of the building. (a) has high uncertainty in moving direction. (b) has high localization error in lateral direction. (c) layout of the building make the localization more accurate.

occurrence of $(i, j)_{th}$ bin and calculated using following equation:

$$P_{normal}(i, j) = \frac{h_{(i,j)}}{\sum_{i=1}^{b} h_{(i,j)}}, \tag{19}$$

where $h_{(i,j)}$ is the value of the $(i, j)_{th}$bin of the normal angle histogram.

If local vicinity has a higher degree of disparity of normal, the histogram bins filled more evenly, therefore the entropy of the normal angle histogram increase. On the other hand, if the $normal\_entropy$ shows lower value, it means that the local vicinity has a low level of scatter, and localization might be erroneous. Therefore localization accuracy has direct relation with $normal\_entropy$ as shown in the followings.

$$localization\ accuracy \propto normal\_entropy\ , \tag{20}$$

Value of $normal\_entropy$ is both related to the environment and abstraction ratio of the map. Figure 5-19 shows the effect of abstraction ratio in the $normal\_entropy$. As shown in 4.0m ND map, the abstraction causes the normals to point to the same direction. In this case localization in the moving direction might have error.

5.4.3. *angular_entropy*

One of the factors which can help the quantification of the layout of the map is $angle\_entropy$. $angle\_entropy$ shows a degree of uniformity of the feature distribution in the space. In order to calculate $angular\_entropy$, the azimuth of all features are calculated. Then, based on azimuth value, angular histogram are filled. Angular histogram is one dimension and has 90 bins. From the angular histogram, $angular\_entropy$ is calculated as follows:

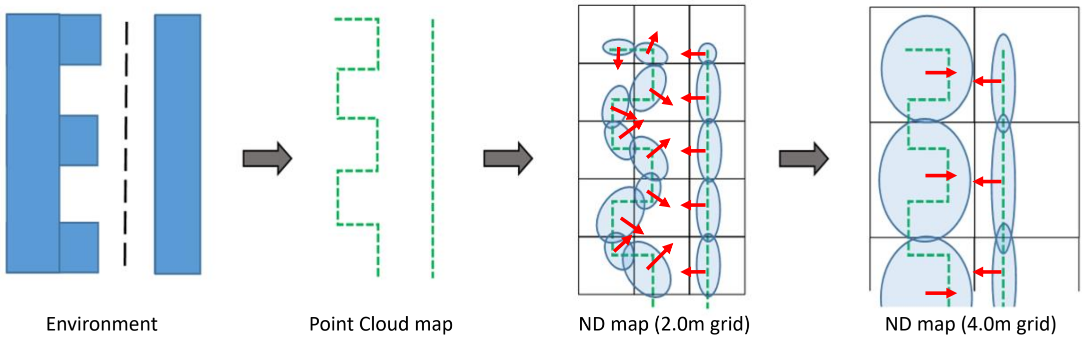


Figure 5-19 Effect of map resolution on the normal of the features. Red arrows are the normal of each featue (ND).

$$H_{angular} = -\sum_{i=1}^{b} P_{angular}(i)\left(log_2 P_{angular}(i)\right) , \tag{21}$$

where b is the number of bins which is 90, and $P_{normal}(i)$ is the probability of occurrence of i_{th} bin and calculated using following equation:

$$P_{angular}(i) = \frac{h_i}{\sum_{i=1}^{b} h_i} , \tag{22}$$

where h_i is the value of the i_{th}bin of the angular histogram.

The more features distribute uniformly around the vehicle, the higher $angular\_entropy$ achieved. Higher $angular\_entropy$ value might have positive effects on the localization accuracy. Abstraction ratio does not affect the $angular\_entropy$.

5.4.4. $r\_average$

$r\_average$ factor is the average of the distance of the features in the local vicinity from point P and calculated as follows:

$$r_{average} = \sum_{i=1}^{m} r_i, \tag{23}$$

where m is the number of features (NDs) in the local vicinity and r_i is the distance to the i_{th} feature from the center of the local vicinity.

5.5. Factor for representation quality

In order to quantify the representation criterion, mahalanobis_distance factor is proposed. This factor shows how much a ND_{map} of the local vicinity was able to preserve the details of the raw point cloud S.

To formulate this, for each point in the raw point cloud of local vicinity S, distance to corresponding ND is calculated. Distance of point to the ND is calculated by Mahalanobis distance as it consider the scaling parameters. Mahalanobis distance d_{mah} of the point x_i to the ND_j is calculated as follows:

$$d_{mah}(x_i) = \sqrt{(x_i - \mu_j)^T {\sum_J}^{-1} (x_i - \mu_j)} , \tag{24}$$

where μ_j and $\sum_j$ are mean and covariance of the nearest ND_j to x_i respectively. Then average of these points for the local vicinity are considered as $mahalanobis\_distance$ factor and calculated as follows:

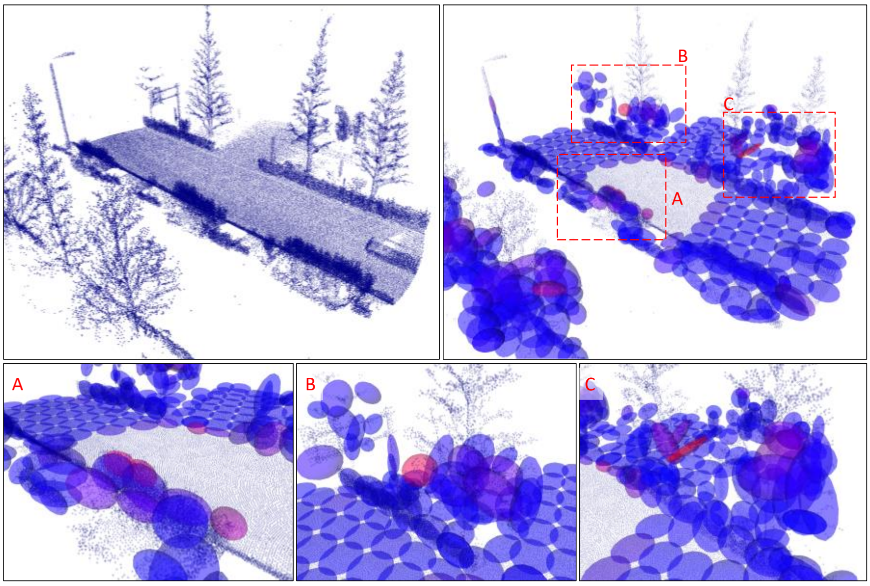


Figure 5-20 Different Mahalanobis distance values for different NDs. Grid size is 2.0m. Blue shows lower mahalanobis distance and red shows higher values. Ground shows low value means the representation quality is high and vegetation and trees have higher Mahalanobis value means the representation quality is low.

$$D_{mahalanobis} = \frac{\sum_{i=1}^{n} d_{mah}(x_i)}{n} \;, \tag{25}$$

where n is the total number of points in the local vicinity of P.

Figure 5-20 shows Mahalanobis distance value for different NDs. Blue shows lower Mahalanobis values and red shows higher values. NDs correspond to ground shows lower Mahalanobis values, means the representation quality is high. However, vegetation and trees have a color near to red. This is because in these points representation quality is low. In fact, for the trees and vegetation, ND with 2.0m grid cannot represent the details.

5.6. Factors for local similarity

5.6.1. $score\_entropy$

To formulate the local similarity of the map in position P, the $score\_entropy$ is introduced.

The score of registration of local vicinity point cloud $\boldsymbol{S}$ on its $\boldsymbol{ND_{map}}$ is calculated as

$$S(\vec{v}) = -\sum_{k=1}^{n} \tilde{p}\left(T(\vec{v}, x_i)\right) \;, \tag{26}$$

where $\tilde{p}$ is the simplified log-likelihood function of the nearest normal distribution to the point x_i in the ND_{map} and $\mathrm{T}(\vec{v}, x_i)$ is the transformation function which transforms x_i with 2D transformation vector $\vec{v}(x, y)$. Here $\vec{v}$ can only transform point x_i in two dimensions, x dimension and y dimension.

It is obvious that if the transformation vector is set to $\vec{v}(0, 0)$, the score function shows maximum value. As shown in Figure 5-21(b) if the point cloud shifted Δ to the right or left the score function decreases as the likelihood of S and ND_{map} are decreases. However, in Figure 5-21(a), as map has repetitive characteristics, by shifting S to the left and right, numerous peaks appear. This peaks usually known as local optimum and make the optimization process stack in a wrong answer. Comparison of the score function $S_a(\vec{v})$ with $S_b(\vec{v})$, which both are two dimension distributions, shows that in the presence of the repetitive objects in the map which is called local similarity here, the score function will distribute more evenly. Therefore if the score function is assumed as a probability distribution, and the entropy of the score function is calculated, $score\_entropy$ in the scenario such as Figure 5-21(a) shows higher value.

In order to calculate the $score\_entropy$, first the score is changed to the probability distribution as follows:

$$P_{score}(\vec{v}) = \frac{s(\vec{v})}{\sum s(\vec{v})}, \tag{27}$$

where $s(\vec{v})$ is the score of registration for $\vec{v}$ and $P_{score}(\vec{v})$ is the corresponding probability. Using (25) $score\_entropy$ is defined as the entropy of the P_{score} as follows:

$$H_{score} = -\sum_{\vec{v}} P_{score}(\vec{v}) log_2\left(P_{score}(\vec{v})\right), \tag{28}$$

where H_{score} is $score\_entropy$.

Range and steps of the transformation vector $\vec{v}$ is defined based on the initial guess requirements and here set to $[-2m, 2m]$ for both direction x and y with steps of $20cm$.

As shown in the following, score entropy has a direct relation to the local similarity criterion.

$$local\ similarity \propto H_{score}, \tag{29}$$

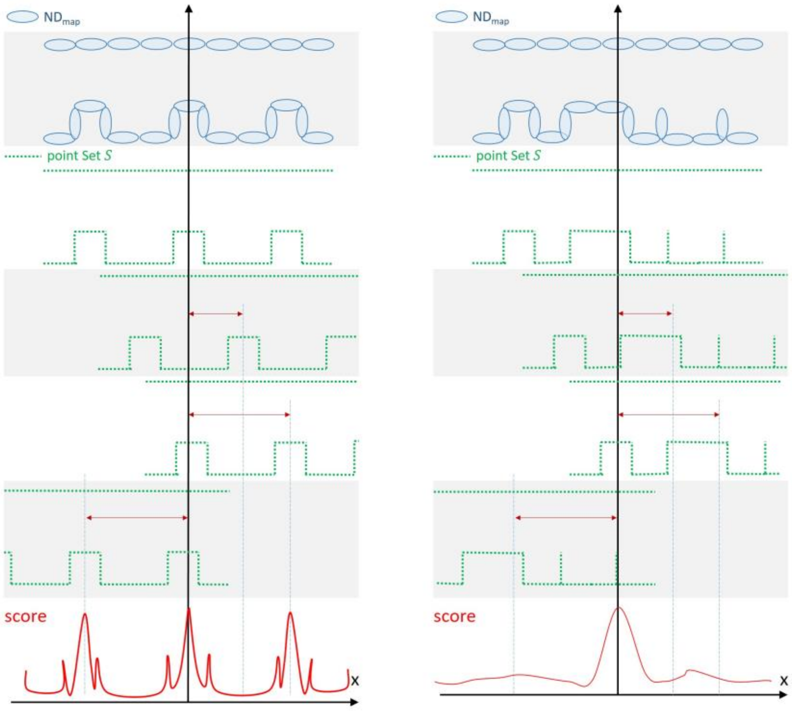


Figure 5-21 Score for two different scenario. Left environment has local similarity and right does not. Left score has more peaks. Score is obtained by moving point cloud (green points) over ND map (blue ellipsoid) and calculating likelihood.

5.6.2. $pfh\_similairty$

In addition to the aforementioned factor, similarity based on Point Feature Histogram (PFH) is considered as another factor to calculate the similarity. A Point Feature Histogram representation is based on the relationships between the points in the k-neighborhood and their estimated surface normals. Simply put, it attempts to capture as best as possible the sampled surface variations by taking into account all the interactions between the directions of the estimated normals. The resultant hyperspace is thus dependent on the quality of the surface normal estimations at each point [82], [83]. In the $pfh\_similarity$ for each ND entity of the map, PFH is calculated based on the other NDs in the vicinity. This PFH represent the characteristics of that ND or in other word, the surrounding environment. Then this histogram is compared to other histogram in the local vicinity. Two approaches for calculating similarity from PFH is followed. First, to calculate the Manhattan distance of the PFH similarity of each factor to other neighbors. And total sum of Manhattan distance is assumed as the similarity value for that local vicinity. If the Manhattan distance is high, it means the PFH of two neighbors are not similar and thus the similarity between them is low. The other approach is to calculate the entropy from PFH. Lower entropy means more similarity of neighbors.

5.6.3. $D\_Battacharyya$

The other approach for calculating the local similarity is to calculate the Battarcharyya distance for each pair of ND entities. Battacharyya can measure the distance between two discrete and continuous probability distribution function. Bhattacharyya distance $D_B(p,q)$ between two distributions $p(x) = N(x|\mu_p, \Sigma_p)$ and $q(x) = N(x|\mu_q, \Sigma_q)$ is defined as:

$$D_B(p,q) = \frac{1}{8}(\mu_p - \mu_q)^T (\frac{\Sigma_p + \Sigma_q}{2})^{-1}(\mu_p - \mu_q) + \frac{1}{2} log\left(\frac{\frac{1}{2}\left|\frac{\Sigma_p + \Sigma_q}{2}\right|}{|\Sigma_p|^{0.5}|\Sigma_q|^{0.5}}\right), \quad (30)$$

Summation of all value from the local vicinity is $D\_battacharyya$.

Chapter 6.
Digital map evaluation based on quantification factors

In this chapter, the effectiveness and contribution of the proposed factors for the map evaluation is investigated by real experiments. First the experimental parameters and methodology is described. Then the effectiveness of the factors are investigated by correlation factor and Principal Component Analysis. Then using the Principal Component Regression (PCR), the localization error is modeled. The results are evaluated with and without similarity criteria to show the contribution of it. Finally, in order to model the error better, simple feedforward ne network is used and the results are shown.

6.1. Experimental methodology and setup

To evaluate the contribution of each factor in the map matching-based localization error, an experiment was conducted in Shinjuku, a dense urban area of central Tokyo, Japan. Figure 6-1 shows the experimental area and paths. Total route of the experiments are 40Km. Streets around Shinjuku is surrounded by skyscrapers, tall buildings, narrow streets, and trees. Localization is performed using VELODYNE’s VLP-16 which has 16 channels. The VLP-16 is placed on the roof of the vehicle with the height of 2.45 m. The sensor setup for the experiments are shown in Figure 6-2. To avoid the scan distortion due to the motion of the vehicle, the vehicle’s velocity was below 2 m/s while the frequency of the laser scanner was set to 20 Hz which limits the distortion in each scan to less than 10 cm. From point cloud map (raw point cloud) the ND map is generated with the various grid sizes. The factors are extracted for sample points and the extracted factors are compared with the localization error for the same sample point. Sample points for the map evaluation are selected along the trajectory with 1.0m intervals. For each sample point, localization error and map factors are calculated. In order to calculate the localization error for each sample point, particles are distributed within the range of 2.0m.

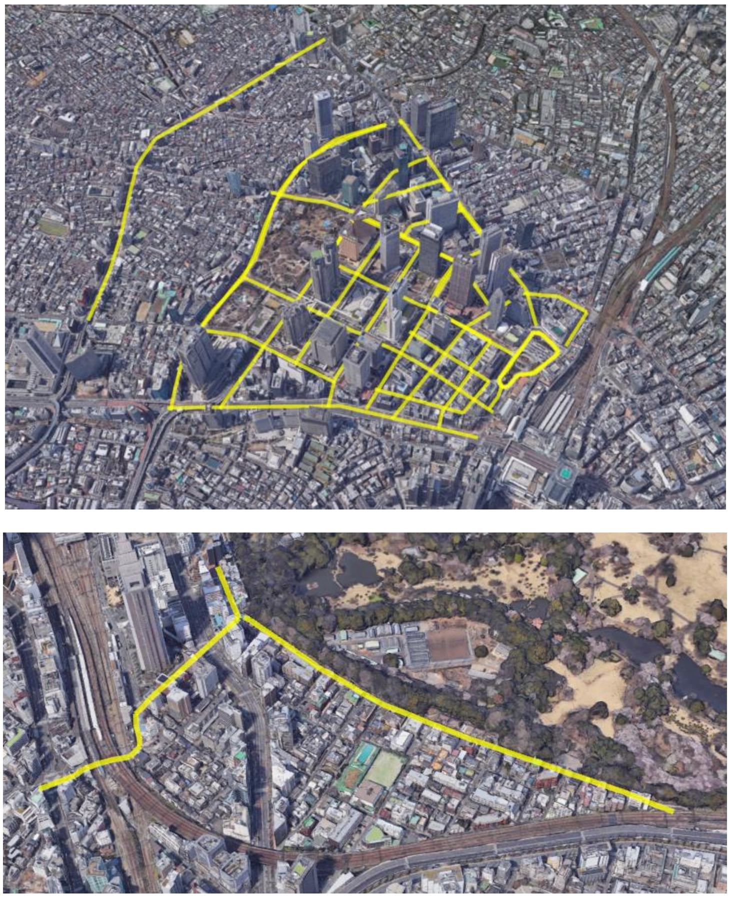


Figure 6-1 path of experiments in Shinjuku

These particles are distributed with and interval of 0.2m from each other in 2D grid. The distribution of particles are shown in Figure 6-3. Localization (map-matching) is performed for each particles and the localization error is calculated. In fact, the particles

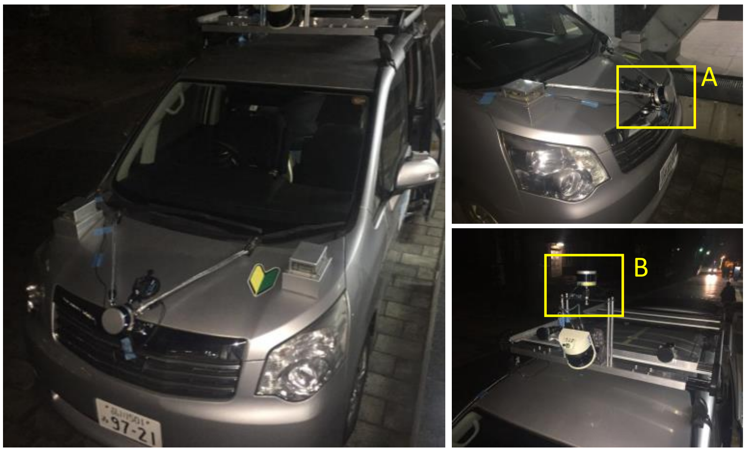


Figure 6-2 Our experimental vehicle. Front Velodyne VLP-16 is used for mapping and it is tilted to scan the environment densely (pith -80°). Top VLP-16 used for localization.

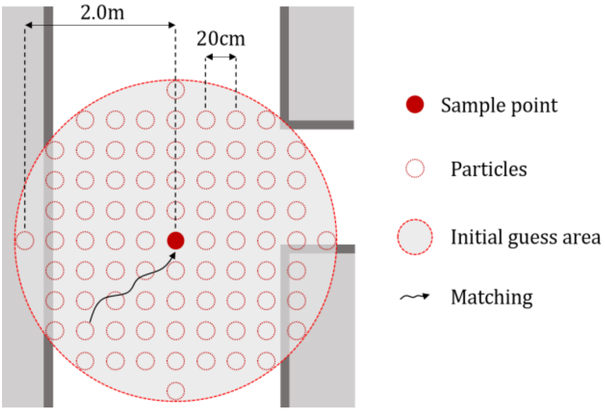


Figure 6-3 sample points and particles which are distributed in the vicinity of the sample point.

can be assumed as different initial guess for map-matching. Using the localization errors of the particles, the mean and maximum error for each sample point are calculated. The maximum error are more important for evaluation however the maximum error are tolerate a lot. Because of this tolerance, modeling is difficult. Therefore, worst 10% error in each particle is assumed as error parameter for map evaluation. Localization error is calculated by comparing with the ground truth. Ground truth is obtained by the 1.0m grid map-matching with two extensions. First, in order to not stack in local minimum, the score of the matching for different particles around the sample point is calculated. The highest one is selected as the one which is most close to ground truth. Then, again in the vicinity of this sample point, particles are distributed with closer intervals and the score of matching for each of the particles are calculated. Highest score is obtained as the ground truth. This ground truth is then visually confirmed. For the same sample points, map factors are calculated as well. Map factors are calculated only based on the ND map in the local vicinity of sample point and input scan is not considered.

6.2. Evaluation and error modeling with PCR

Figure 6-4 shows the comparison of four defined factors with 3D maximum error for 4.0m ND map (grid size in this ND map is 4.0m). These factors are DM_ratio, FDOP, score_entropy (H_{score}), and normal_entropy (H_{normal}). Normal entropy is calculated for 8x8 bins. In region (a) and (c) where the score_entropy is lower compared to the other parts, the localization error is higher. The same thing happens for most of the peaks of the maximum error. However, in region (b) the score_entropy stayed high while the localization error was increased. While the similarity entropy failed to describe the maximum localization error in region (b), the normal entropy factor clearly describes the error source.

Figure 6-5, Figure 6-7, Figure 6-8 shows the map evaluation factors compared to mean and maximum error for ND map with 2.0m grid size of PATH #II of the experimental area (Figure 6-6). Figure 6-5 shows the factors for feature sufficiency criterion. The first and second figures show the maximum and mean error of the self-

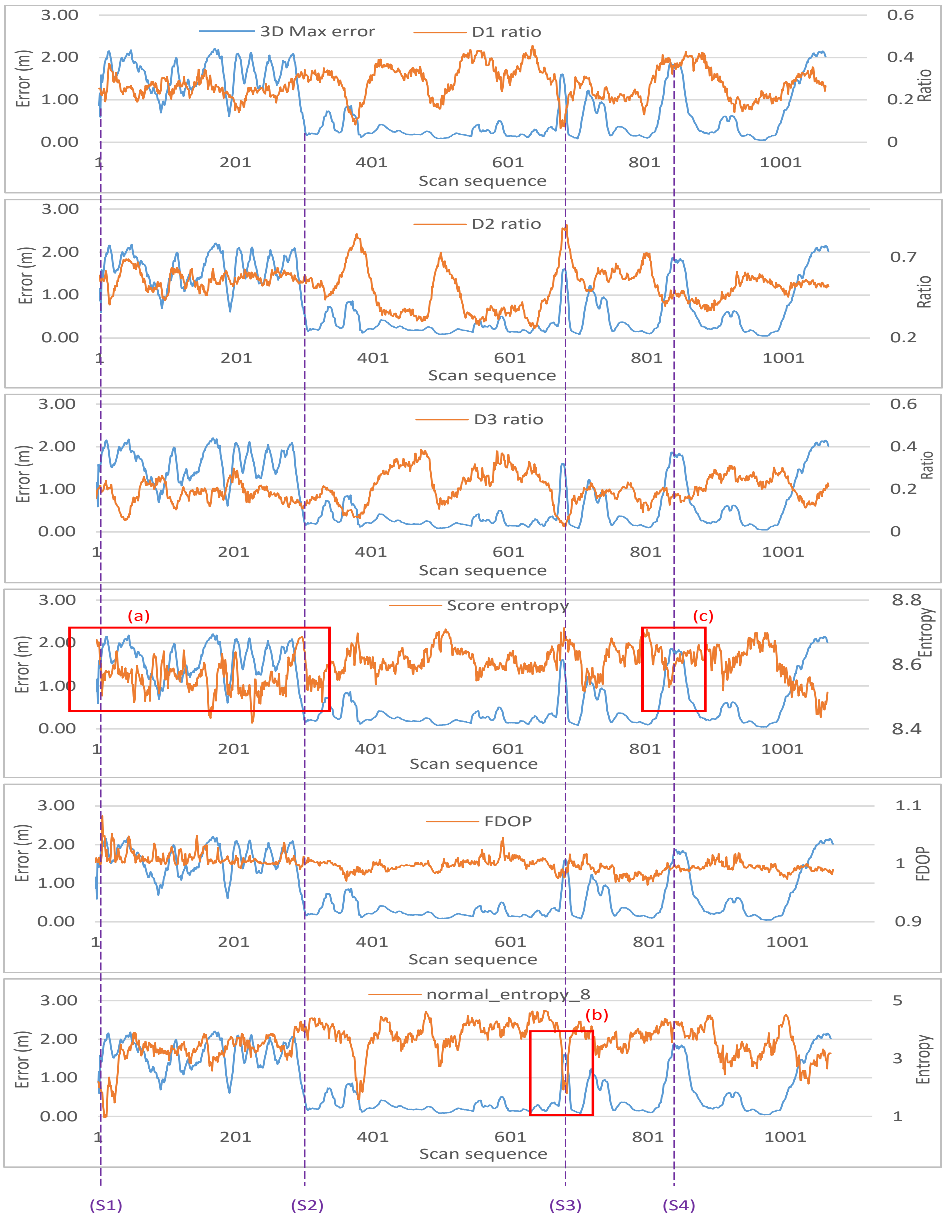


Figure 6-4 Comparison of different factors with 3D maximum error. For each figure, blue graph is 3D maximum error and orange graph is values of factors. Normal entropy factor is calculated for 8×8 bins.

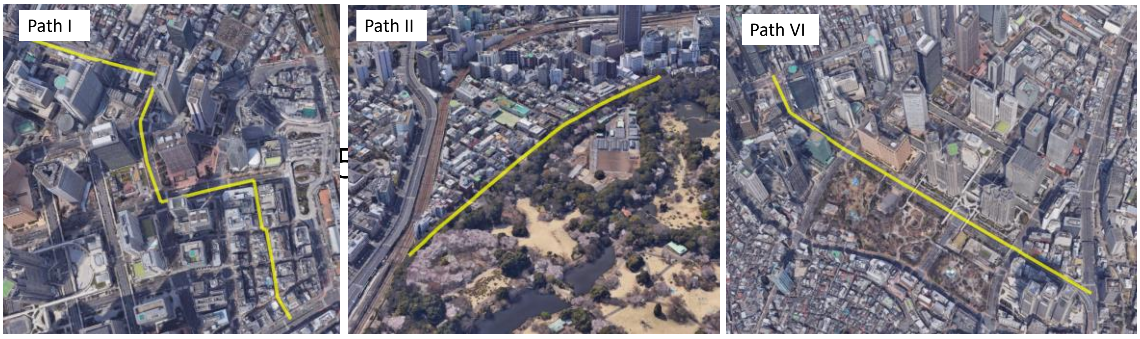


Figure 6-6 Path #I, #II, and #VI of the experimental area.

localization respectively. In B-C and D-E periods, localization shows higher accuracy. However, in A-B and C-D big error appeared. These errors came from many factors which can be assessed from Figure 6-5, Figure 6-7, and Figure 6-8. Figure 6-5 shows that the $occupancy\_ratio$ and $feature\_cnt$ is highly related to the mean and max error. In A-B and C-D where error exist, these two values are lower. Among the Dm_ratio factor ($\forall m \in [1,3]$), $D3\_ratio$ contribute more to the localization accuracy. Whenever the ratio of $D3$ features lowered the localization error increased.

Localization errors and factors related to the Layout criterion for a 4.0m grid of path

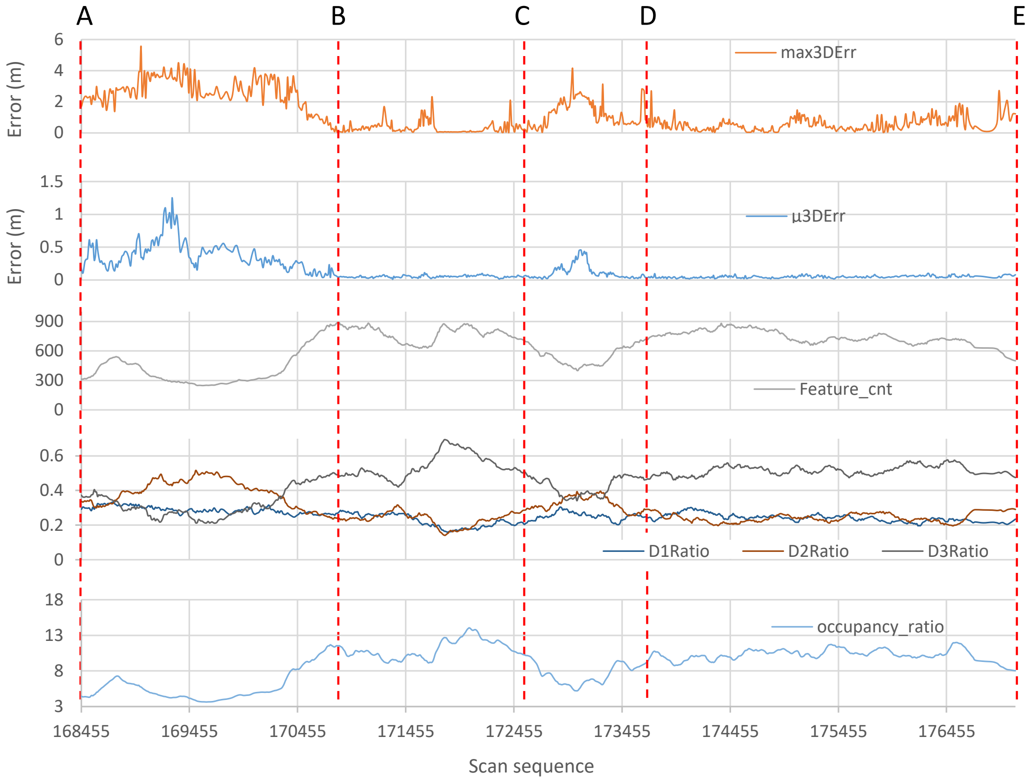


Figure 6-5 Factors related to the feature sufficiency for path II (2.0m ND) compared to mean and max error of the same path.

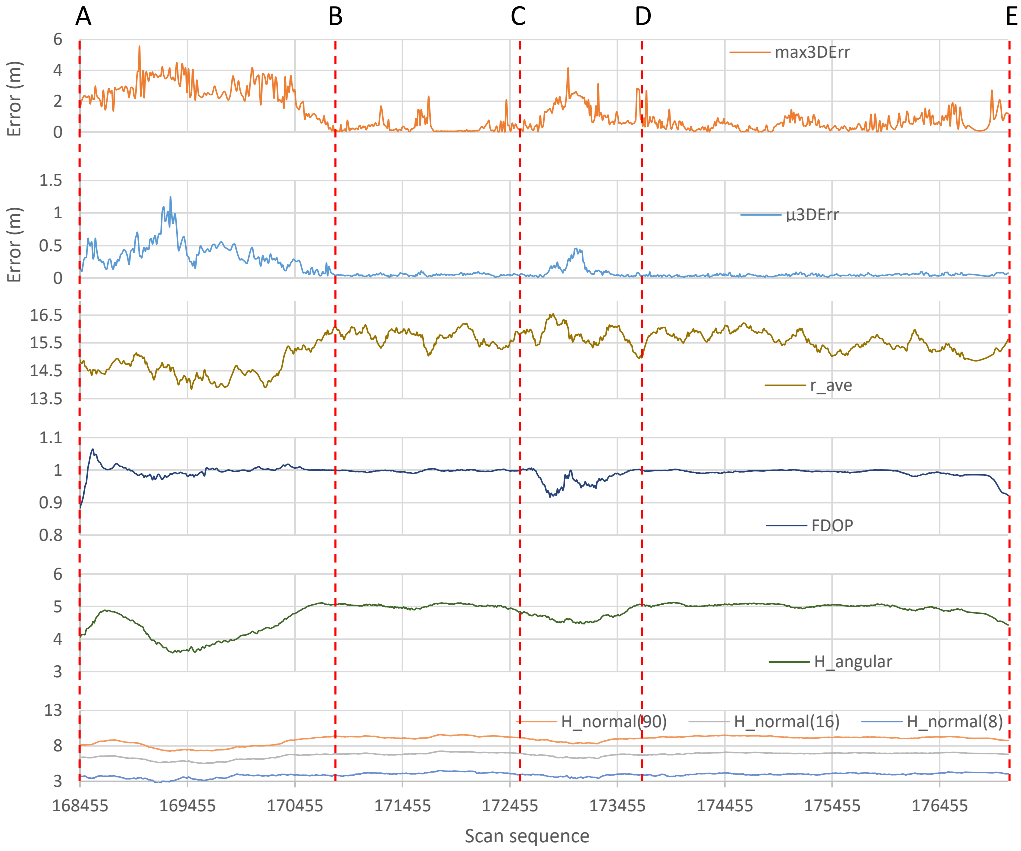


Figure 6-7 Factors related to the layout for path II (2.0m ND) compared to mean and max error of the same path.

II is shown in Figure 6-7. A-B period of Figure 6-7 shows, *r_ave* and

TABLE 6-1 Correlation of factors related to the layout, local similarity, and representation quality with each other and mean and max error for path II (2.0m ND).

criteria	Layout						Representation quality	Local similarity	Localization Error	
factors	r_{ave}	$FDOP$	$H_{angular}$	H_{normal} (90)	H_{normal} (16)	H_{normal} (8)	D_{mah}	H_{score}	μ_{3DErr}	max_{3DErr}
r_{ave}	**1.000**	-0.190	0.684	0.727	0.654	0.463	-0.372	0.580	-0.631	-0.674
$FDOP$	-0.190	**1.000**	0.290	0.222	0.210	0.204	0.286	-0.032	-0.057	-0.134
$H_{angular}$	0.684	0.290	**1.000**	0.972	0.927	0.745	-0.143	0.640	-0.825	-0.809
H_{normal} (90)	0.727	0.222	0.972	**1.000**	0.971	0.811	-0.189	0.700	-0.836	-0.844
H_{normal} (16)	0.654	0.210	0.927	0.971	**1.000**	0.883	-0.177	0.696	-0.824	-0.827
H_{normal} (8)	0.463	0.204	0.745	0.811	0.883	**1.000**	0.022	0.544	-0.763	-0.717
$D_{mahalanobis}$	-0.372	0.286	-0.143	-0.189	-0.177	0.022	**1.000**	-0.421	0.026	0.164
H_{score}	0.580	-0.032	0.640	0.700	0.696	0.544	-0.421	**1.000**	-0.592	-0.654
$Feature_{cnt}$	0.712	0.266	0.923	0.960	0.902	0.727	-0.149	0.646	-0.764	-0.843
$D1_{cnt}$	0.612	0.256	0.811	0.804	0.686	0.441	-0.059	0.486	-0.624	-0.672
$D2_{cnt}$	0.555	0.145	0.631	0.588	0.452	0.238	-0.141	0.301	-0.517	-0.515
$D3_{cnt}$	0.667	0.254	0.878	0.939	0.926	0.807	-0.157	0.668	-0.743	-0.837
$D1_{ratio}$	-0.406	-0.042	-0.460	-0.542	-0.637	-0.700	0.179	-0.470	0.509	0.557
$D2_{ratio}$	-0.655	-0.217	-0.906	-0.950	-0.961	-0.824	0.179	-0.703	0.761	0.808
$D3_{ratio}$	0.637	0.180	0.847	0.908	0.949	0.866	-0.197	0.696	-0.753	-0.805
$occup_{ratio}$	0.641	0.256	0.874	0.924	0.921	0.807	-0.190	0.617	-0.752	-0.831

1.0

-1.0

Green cells show high positive relation and red cells show high negative relation. Yellow cells has no relation.

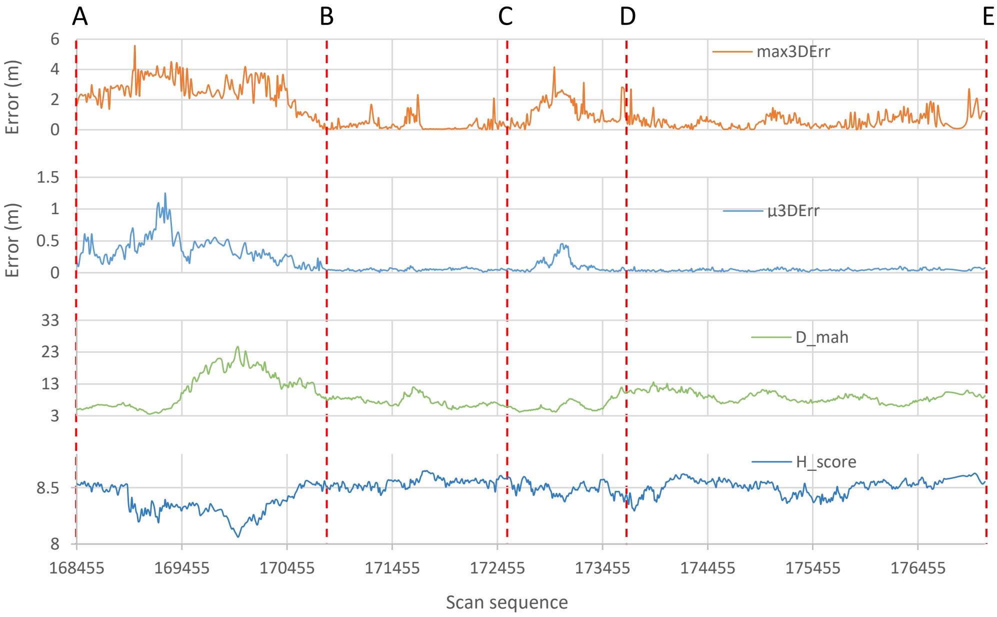


Figure 6-8 Factors related to the representation quality and local similarity for path II (2.0m ND) compared to mean and max error of the same path.

$angular\_entropy$ react better than other factors. It is shown that if the r-average is higher, localization become more accurate. In C-D, $FDOP$, $angular\_entropy$, and $normal\_entropy$ react well. For both A-B and C-D, $normal\_entropy$ reacts well. In Figure 6-7 $normal\_entropy$ is calculated for 90×90, 16×16bins, and $8 \times$

TABLE 6-2 Correlation of factors related to the feature sufficiency with each other and mean and max error for path II (2.0m ND).

criteria	Feature sufficiency								Localization Error	
$factors$	$Feature_{cnt}$	$D1_{cnt}$	$D2_{cnt}$	$D3_{cnt}$	$D1_{ratio}$	$D2_{ratio}$	$D3_{ratio}$	$occup_{ratio}$	μ_{3DErr}	max_{3DErr}
$Feature_{cnt}$	**1.000**	0.866	0.639	0.964	-0.504	-0.912	0.866	0.932	-0.764	-0.843
$D1_{cnt}$	0.866	**1.000**	0.722	0.721	-0.017	-0.716	0.552	0.679	-0.624	-0.672
$D2_{cnt}$	0.639	0.722	**1.000**	0.444	-0.081	-0.344	0.290	0.518	-0.517	-0.515
$D3_{cnt}$	0.964	0.721	0.444	**1.000**	-0.670	-0.950	0.952	0.955	-0.743	-0.837
$D1_{ratio}$	-0.504	-0.017	-0.081	-0.670	**1.000**	0.586	-0.787	-0.690	0.509	0.557
$D2_{ratio}$	-0.912	-0.716	-0.344	-0.950	0.586	**1.000**	-0.961	-0.902	0.761	0.808
$D3_{ratio}$	0.866	0.552	0.290	0.952	-0.787	-0.961	**1.000**	0.922	-0.753	-0.805
$occup_{ratio}$	0.932	0.679	0.518	0.955	-0.690	-0.902	0.922	**1.000**	-0.752	-0.831
r_{ave}	0.712	0.612	0.555	0.667	-0.406	-0.655	0.637	0.641	-0.631	-0.674
$FDOP$	0.266	0.256	0.145	0.254	-0.042	-0.217	0.180	0.256	-0.057	-0.134
$H_{angular}$	0.923	0.811	0.631	0.878	-0.460	-0.906	0.847	0.874	-0.825	-0.809
H_{normal} (90)	0.960	0.804	0.588	0.939	-0.542	-0.950	0.908	0.924	-0.836	-0.844
H_{normal} (16)	0.902	0.686	0.452	0.926	-0.637	-0.961	0.949	0.921	-0.824	-0.827
H_{normal} (8)	0.727	0.441	0.238	0.807	-0.700	-0.824	0.866	0.807	-0.763	-0.717
$D_{mahalanobis}$	-0.149	-0.059	-0.141	-0.157	0.179	0.179	-0.197	-0.190	0.026	0.164
H_{score}	0.646	0.486	0.301	0.668	-0.470	-0.703	0.696	0.617	-0.592	-0.654

1.0

-1.0

Green cells show high positive relation and red cells show high negative relation. Yellow cells has no relation.

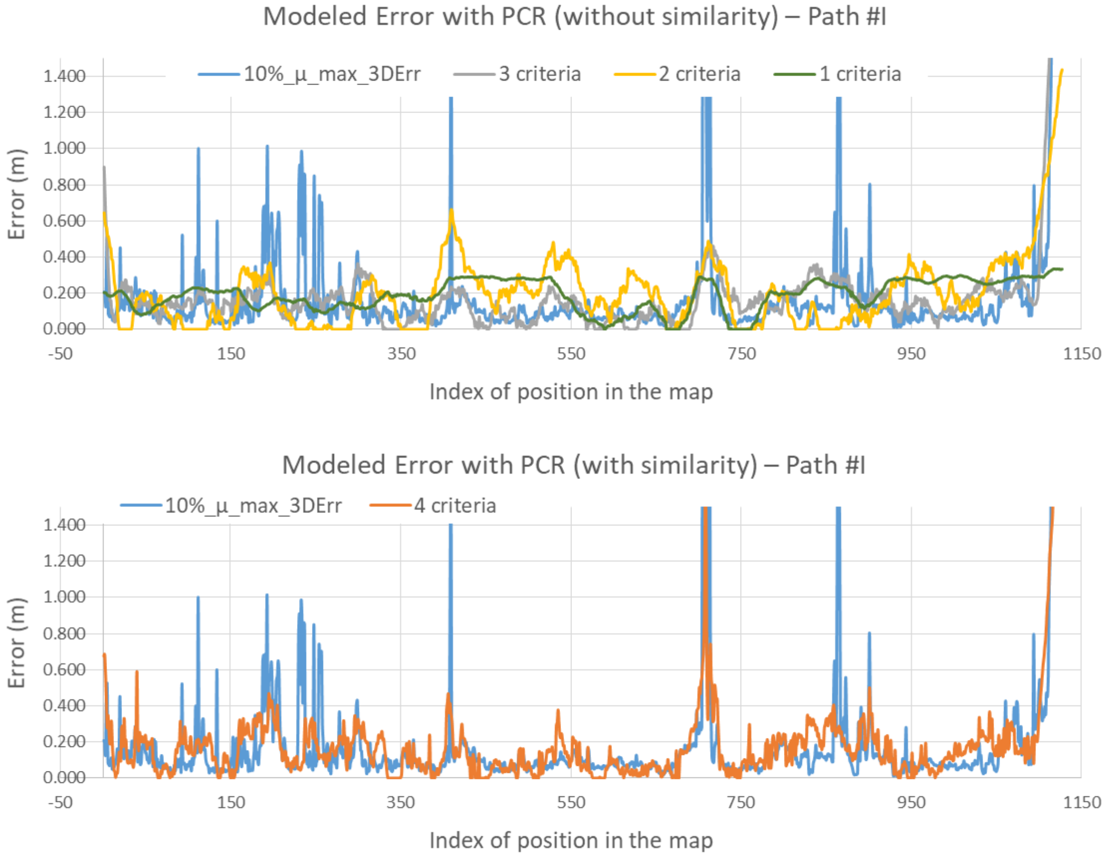
8bins. Among these three 90×90bins can show error model more precisely.

Localization errors and factors related to both local similarity and representation quality criteria for a 2.0m grid of path II is shown in Figure 6-8. Errors are made of several factors, and each of factors reacts in a specific situation. In other words, each of the factors can model something that others cannot. Thus, for modeling the error, all of them are essential. Factors can overlap in some situations.

For example, in Figure 6-8, both $occupancy\_ratio$ and $feature\_cnt$ follows the similar trend. However, it is not always and at the same time they show different characteristics of the map as well. TABLE 6-2 shows a correlation of the factors related to the feature sufficiency with other factors and also with the mean and max error.

Likewise, TABLE 6-1 shows correlations for factors related to the layout, local similarity and representation quality. Correlation of the factors with errors can be assumed as effectiveness or contribution of each factor to the error. In other words, the higher the correlation value with error model, the more that factor contribute to the error. In TABLE 6-2 and TABLE 6-1, a value near to 1.0 (green cells) shows two variables are act highly related to each other and values near to -1.0 (red cells) shows they are act nearly opposite of each other. Values near the 0 (yellow cells) means they are not related at all. In TABLE 6-2 $occupancy\_ratio$ and $feature\_count$ has 0.93 of correlation. High correlation value could be expected from Figure 6-5 as well. Factors related to the feature sufficiency (TABLE 6-2), has high correlation values with localization error. Among them, $feature\_count$ has the highest correlation with 0.843. Contribution of each factor is different in mean and maximum error. Correlation of $D2\_ratio$ with error (0.761 for mean, 0.808 for max) shows that if the proportion of $D2$ increase, the localization increase too. Among the features in TABLE 6-1, $FDOP$ and $mahalanobis\_distance$ has lower correlation to other factors. Therefore these two factors can evaluate some characteristics of the map that cannot be evaluated by others and make them a key factor for modeling the error. On the other hand, $FDOP$ and $mahalanobis\_distance$ factors has low correlation with the localization error (TABLE 6-1). This is because some of the map criteria does not appear frequently, thus, corresponding factors has lower correlation to the error. One of the other reason is that the representation quality in the 2.0m grid size is not bad. Thus the error is not related to the value of representation quality. Low correlation in TABLE 6-2 and TABLE 6-1 do not necessarily mean the inability of the factors for map evaluation.

In order to model the localization error based on the afformentioned factors, principal component regression (PCR) is used. PCR applied to a set of factors as explanatory variables and localization error as a response. Here, in order to show the importance of the similarity, the error modeling is performed with and without similarity. PCR are applied to all data. The results are shown for three paths #I, #II, and #VI in the Figure 6-6 and total path (Figure 6-1).


Modeled Error with PCR (without similarity) – Path #I

Modeled Error with PCR (with similarity) – Path #I

Figure 6-9 Modeled error with PCR for Path #I with (bottom) and without similarity (top).

In order to show how much the proposed factors could model the error, r-squared (R^2) and root mean squared error (RMSE) are used. In addition to these parameters, ratio of predicted error less than 10cm ~ 20cm are shown in the Tables. This ratio means what percent of the sample points can be evaluated with error less than 10cm. Figure 6-9 shows the modeled error using PCR for Path #I. The results are divided into two parts. Modeled error without using the fourth criteria which is local similarity criterion (top) and the

Table 6-3 Modeled error goodness of fit for Path #I

	R^2	RMSE	predErr < 10cm	predErr < 15cm	predErr < 20cm	predErr > 25cm
Represent Quality	0.046	0.399	47.6%	65.2%	79.9%	7.2%
+Feature Criteria	0.269	0.350	40.9%	60.2%	72.5%	18.5%
+Layout Criteria	0.664	0.238	66.2%	83.3%	89.7%	7.7%
+Local Similarity	**0.747**	**0.207**	**70.7%**	**83.2%**	**88.9%**	**7.5%**

results using all criteria (bottom). In Figure 6-9(top), the results are shown by adding each criteria into the model. Graph for 1 criterion only uses the factors for representation quality (green). In graph with 2 criteria, feature sufficiency is added to the model (yellow) and in the graph with 3 criteria, layout is added to the model. In Figure 6-9 (Bottom) similarity criteria added to the model and make the graph fit better to the error model. From Figure 6-9, it is obvious that by adding more evaluation criteria to the model, goodness of fit increased. The results of good ness of fit in Table 6-3 confirm this. The

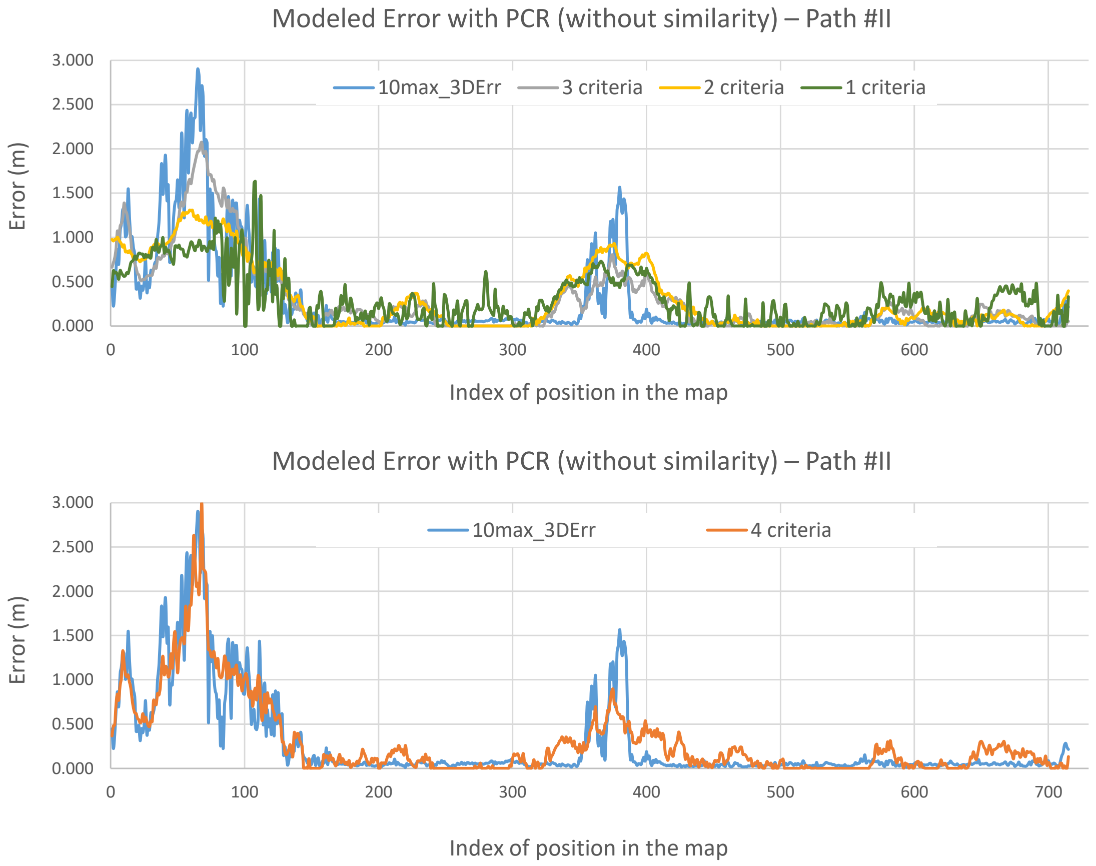


Figure 6-10 Modeled error with PCR for Path #II with (bottom) and without similarity (top).

Table 6-4 Modeled error goodness of fit for Path #II

	R^2	RMSE	predErr < 10cm	predErr < 15cm	predErr < 20cm	predErr > 25cm
Represent Quality	0.404	0.391	37.5%	48.1%	57.5%	34.4%
+Feature Criteria	0.621	0.312	58.6%	66.6%	72.2%	23.5%
+Layout Criteria	0.752	0.253	59.3%	69.8%	78.5%	16.1%
+Local Similarity	**0.819**	**0.218**	**60.1%**	**70.9%**	**78.5%**	**16.1%**

modeled error are shown for the Path #II (Figure 6-10) and Path #VI (Figure 6-11) as well. Also, Table 6-4 and Table 6-5 shows the goodness of fit for Path #II and Path #VI respectively. From the tables, it is obvious that by adding the criteria, better model achieved. The improvement of each path is different from each other. For example, in Path #I, representation quality can model the error in 47.6% points of the map, while in Path #II it is 37.5%. This is because the effect of the representation quality in Path #II is lower. Table 6-6 shows the goodness of fit parameter for whole experimental area. The results show in about 63% of the cases the localization error could be modeled by four

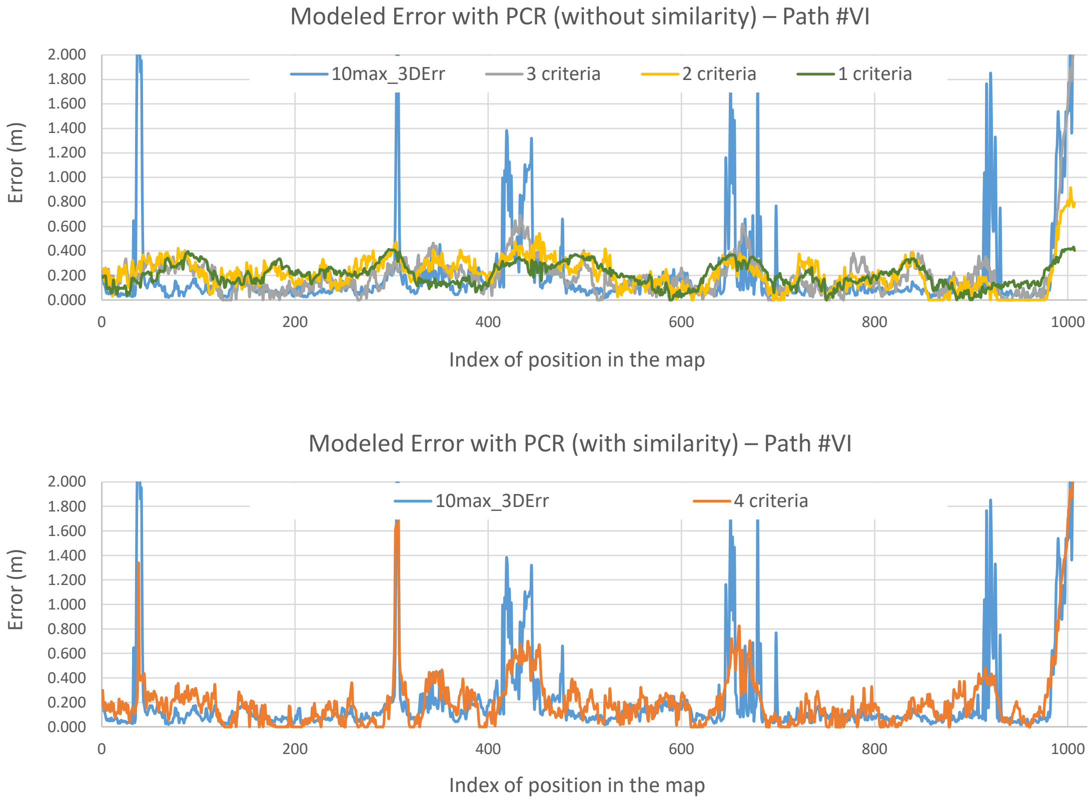


Figure 6-11 Modeled error with PCR for Path #VI with (bottom) and without similarity (top).

Table 6-5 Modeled error goodness of fit for Path #VI

	R^2	RMSE	predErr < 10cm	predErr < 15cm	predErr < 20cm	predErr > 25cm
Represent Quality	0.072	0.342	41.3%	61.0%	77.2%	13.9%
+Feature Criteria	0.185	0.321	46.4%	62.7%	76.0%	15.5%
+Layout Criteria	0.387	0.279	57.9%	73.4%	82.8%	10.8%
+Local Similarity	**0.543**	**0.242**	**59.8%**	**75.8%**	**84.6%**	**10.6%**

Table 6-6 Modeled error goodness of fit for whole area

	R^2	RMSE	predErr < 10cm	predErr < 15cm	predErr < 20cm	predErr > 25cm
Represent Quality	0.064	0.406	35.7%	54.8%	68.3%	20.4%
+Feature Criteria	0.220	0.371	42.1%	55.3%	67.1%	22.5%
+Layout Criteria	0.567	0.276	58.6%	75.0%	82.2%	12.6%
+Local Similarity	**0.666**	**0.243**	**63.3%**	**74.3%**	**82.1%**	**12.8%**

map criteria and using PCR. However this value is not satisfactory. In fact, as the relation of the factors with each other is not clear, it is difficult to formulate it. Thus, we use a simple feedforward neural network for modeling the error more precisely.

6.3. Error modeling using neural network

In order to model the error more precisely, in this subchapter a simple feedforward neural network is used. In fact the connection and relation between the map factors are very difficult to model using the PCR. Thus, the determination of the relation of the factors and weights are done by the learning method. In this work, we have used a very simple feedforward neural network for classification. The input size of vectors correspond to the number of factors extracted from map and the label is the error of the path. As the error more than 1.0m are not acceptable, error more tha n 1.0m are all set to 1.0m. By doing this, the output of the network are become 0 to 1.0 which suit to the classic classification problem. In fact, here only the structure of classification is used. The output of 0 to 1.0 is assumed as error of the position of the map. Input of the network has 16 neurons which is the number of factors. The number of hidden layer is 3 and each hidden layer has 100 of neurons. Total number of data are around 20000 which is shown in the Figure 6-1. Here the data is referred to the set of factors extracted from each sample

Table 6-7 Prediction success rate and false negative rate for whole path.

Prediction Success rate (<10cm) %	Prediction Success rate (<15cm) %	Prediction Success rate (<20cm) %
78.2%	**87.3 %**	**91.8%**

position in the map and actual localization error of the map is assumed as label. In order to show the modeled data for each path, like the one for PCR, each path are excluded from the data, and the network is trained. Later, using the trained model, error for each

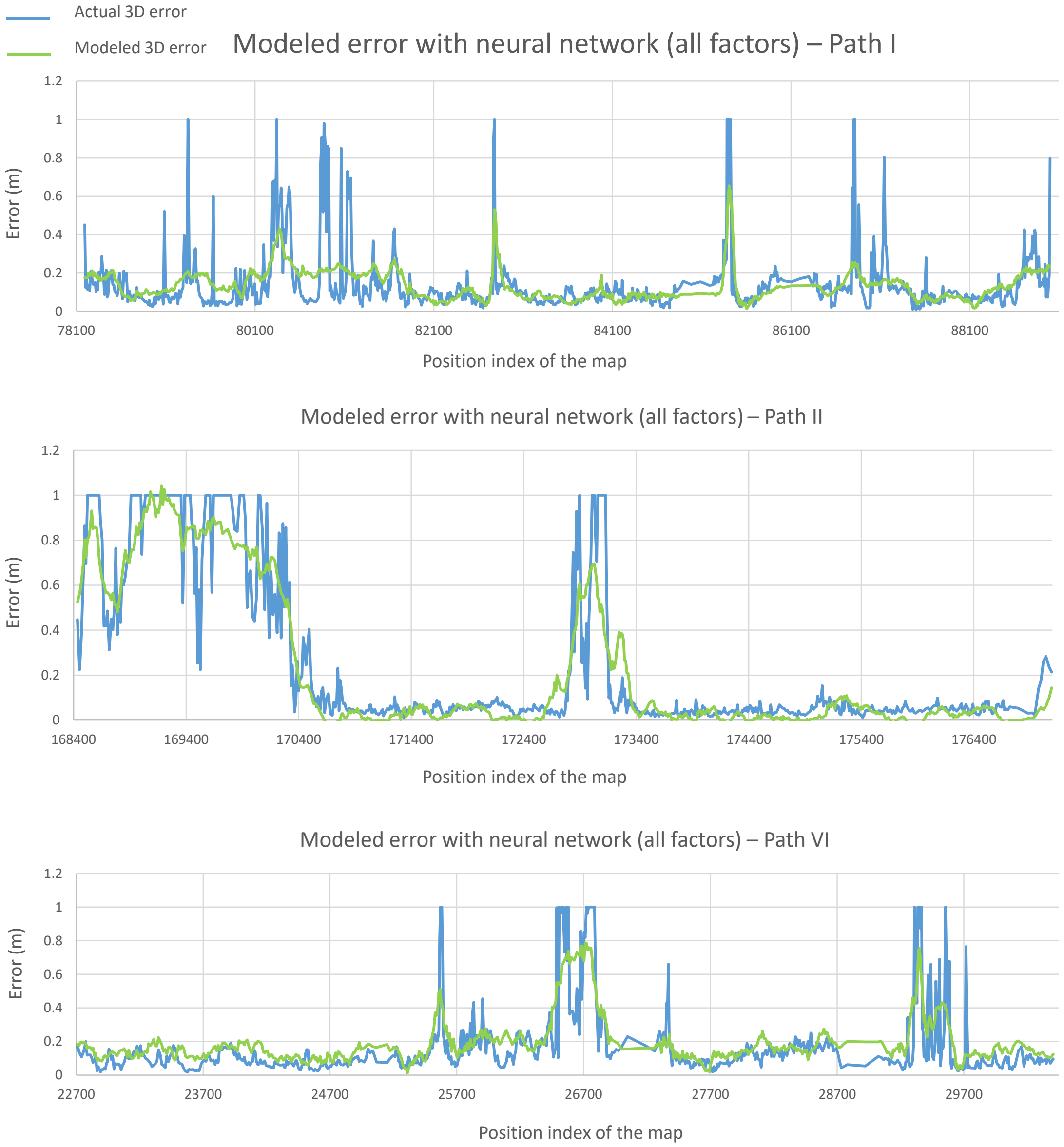


Figure 6-12 Modeled error (green) vs actual error (blue) for the path #I, #II, #VI.

Table 6-8 Prediction success rate and false negative rate for path #I, #II, and #VI.

	Prediction Success rate (<10cm) %	Prediction Success rate (<15cm) %	Prediction Success rate (<20cm) %	Prediction fail rate (>25cm) %	FN rate ($\frac{FN}{TP+FN}$)
PATH #I	82.9%	91.3%	94.6%	4.4%	56.0%(19.6%*)
PATH #II	78.8%	85.0%	90.0%	6.1%	4.26%
PATH #VI	72.1%	83.0%	88.9%	8.25%	26.3% (12.6%*)

*In the case the cut off set to 20cm instead of 25cm

path is modeled. Figure 6-12 shows the modeled error using the learning method. These figures show that the model fitted the actual error better than PCR. The goodness of fit characteristics are shown for each of the path in Table 6-8. Prediction success rate for less than 10cm in the tab les means, if the difference of actual error and predicted error is less than 10cm, then the prediction is assumed as success. For path I, prediction success rate for less than 10cm is 82.9% which is good. Prediction success fail rate for more than 25cm means the prediction and actual error difference is more than 25cm. The rate of fail is 4.4% for path I. The False negative rate which shows the negative prediction over all negative samples are around 4.26% which is quite good. Here, negative means the error is not more than 25cm. as already mentioned in the chapter II about localization, required localization accuracy assumed to be 25cm in this work.

Chapter 7.
Adaptive refinement of map and localization parameters based on map factors

In Chapter 6, map evaluation factors are used to model the error for different position of the map. If error model is available, some of the parameters for localization can be defined beforehand. These parameters are resolution of the map and range of laser scanner. In this chapter, this prediction is used for determination of the abstraction ratio of the map and laser range of the input scan adaptively.

7.1. Adaptive refinement of map resolution

In NDT, map space is divided into the static grids, and for each grid, ND of the points are calculated. Usually, grid size (resolution) of the map is determined empirically and depends on the environment. Bigger grid size has lower map resolution, and smaller grid size has higher resolution. Basically, lower resolution are more favorable because of its data size. In fact, lower resolution has more abstraction ratio and can reduce the map size. However, if the resolution is low, many details of the environment are ignored, and the localization accuracy degrades. In other words, in the bigger grid sizes, some of the vital information for the self-localization are lost and as results error occurs. This information loss and localization error are different from place to place on the map. If the self-localization ability of the map for specific position and resolution can be evaluated with some method, then the grid size of that position can be determined adaptively based on the evaluation.

In this subchapter, localization ability of the map in a specific position for each resolution is evaluated with factors mentioned in Chapter 5. Using these factors, for each position and resolution of the map, localization error are modeled. Based on the modeled error, for each position of the map, (lower resolution) most significant grid size which can preserve required localization accuracy is selected. The adaptive abstraction is evaluated

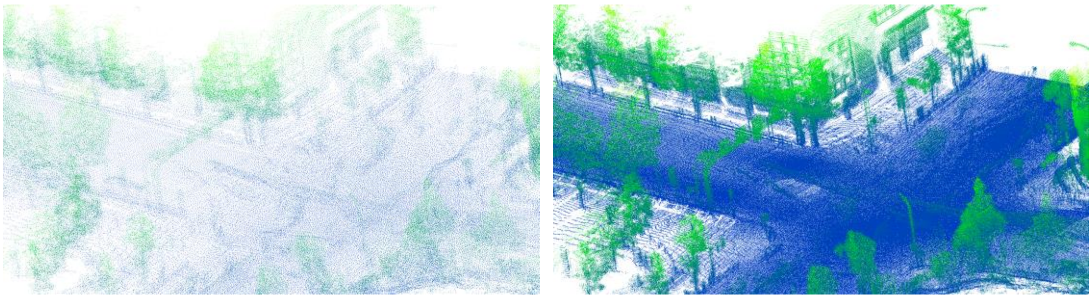


Figure 7-1 Different resolution of the point cloud. Right image is the high density with original resolution. The left image is down-sampled by 30cm grids.

for two required localization accuracy, 0.25m and 0.5m. It is worth to mention that, still there is no global standard for required accuracy of self-localization among the companies; however, Cross-ministerial Strategic Innovation Promotion Program (SIP) Japan have announced in its recent report that the required accuracy of the high definition (HD) map for autonomous driving should be 0.25m. This figure comes from the tier width and satellite image resolution. Based on this figure, the localization accuracy of the autonomous vehicles needs to get close to 0.25m as well.

7.1.1. Effect of map resolution to localization error

Most of the digital map formats such as point cloud map, ND map, Vector map and planar surface map has a kind of abstraction inside it. Otherwise, use of point cloud map itself for localization is very infeasible because of its size. Here, the term resolution is defined for digital map to show the rate of abstraction. Resolution can be used for every map format. In the case of Point cloud map, the term resolution of the map correspond to the grid size which is used for down-sampling the point cloud map. Usually, Point cloud map are down-sampled beforehand due to its huge size. In Figure 7-1 different resolution of the point cloud map is shown.

In the case of ND map, the term resolution refer to grid size that we use to discretize the point cloud before making the NDs. If the smaller grid size used, means the resolution is higher, then more details of the environment can be represented using the NDs. If the bigger grid size is used the resolution become lower and thus less details of the environment represented in the map. The map resolution has direct relation with the map size. Different level of ND resolution are shown in Figure 7-2.

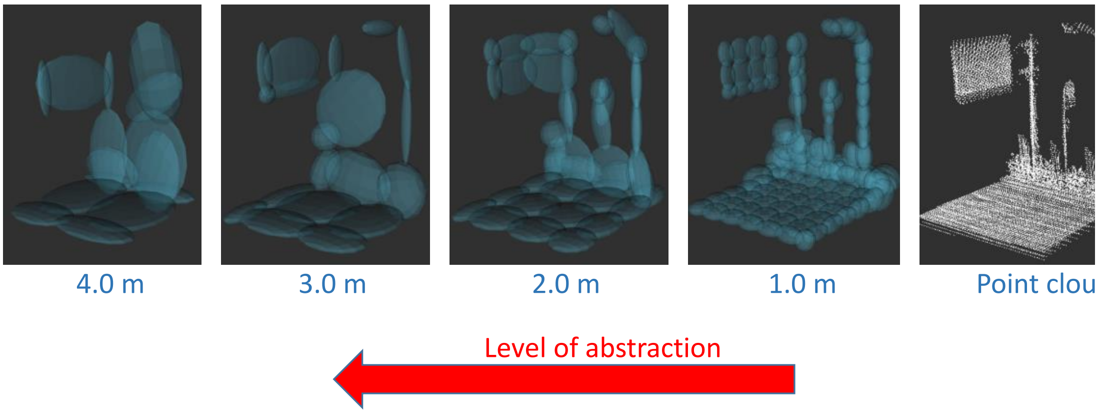


Figure 7-2 Different resolution for ND map. For the ND map, grid size can adjust the resolution.

In the case of multilayered vector map format, the resolution means how much details of the building are appeared in the vector map. By manipulating the line extraction parameters we can change the amount of details that are represented by the map. As shown in the Figure 7-3 for the corresponding point cloud we can extract line segment in two ways. The one with lower resolution (Figure 7-3(right)) and the one with higher resolution (Figure 7-3(left)). As we can see the higher resolution represent more detail of the building (Figure 7-3(left)).

Same concept can be defined for the planar surface map. By manipulating the RANSAC plane extraction parameters, the plane is extracted with more detail which means with higher accuracy. In Figure 7-4 different plane extraction parameter with different resolution are shown.

In general map resolution show how much detail of the environment are represented by the map and has direct relationship with map size. In terms of map size, lower

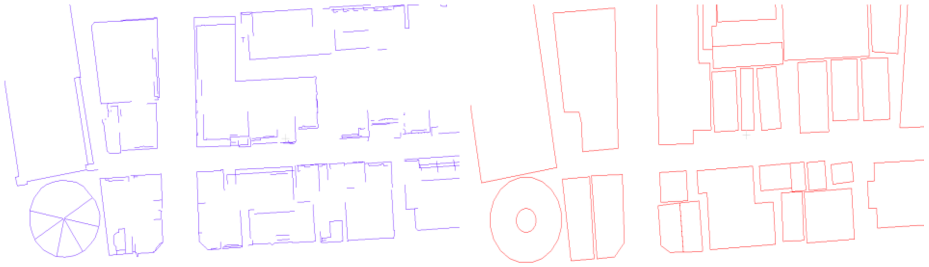


Figure 7-3 Different resolution for the vector map. The vector map resolution can be adjusted using line extraction parameters.

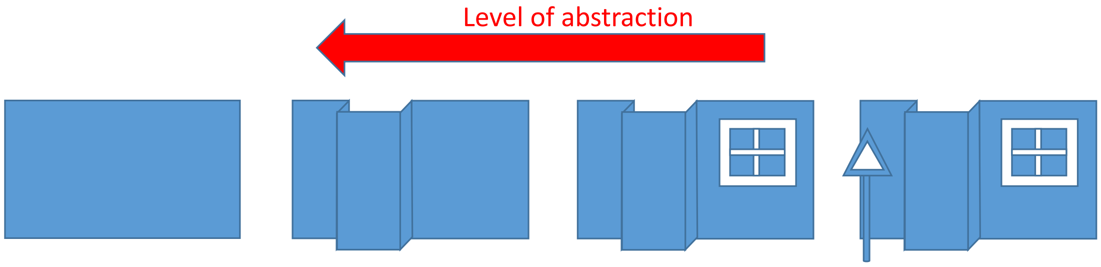


Figure 7-4 Different resolution of plane extraction algorithm. If we set the parameter of plane extraction to very high value, then we have very coarse plane as shown in the left.

resolution is more desi rable. However in terms of self-localization accuracy, more details of the environment can be used for better accuracy. This fact already mentioned in Chapter 3 where we introduced the concept of multilayered vector map. However higher resolution has several disadvantages. These disadvantages are as follows:

- Computation time

  In general, in map-based techniques, for each points in the laser scanner, the corresponding entity is found. If the number of entity is high, then the search time will increase. Thus total matching time increase.

- Intensive memory usage

  More resolution means more entity in the map, thus the memory size increase. In grid-based map, the map size increase logarithmic. If the size of grid become half then the size of memory increase 8 times.

- Data loss

  This will happen in NDT based map matching techniques. Usually for making ND for each cell we require 5 points. In some part of the map, if the resolution of the map is high (grid size is low) then 5 points cannot be found. Thus, in that part we will not have data.

The details of the environment can help the self-localization accuracy in some occasions. For example, In the Figure 7-5 two buildings are shown. Figure 7-5(a) shows a low resolution of the building walls, thus the detail texture of the building is not appeared. In this case the lateral accuracy is high due to presence of two walls. However, for

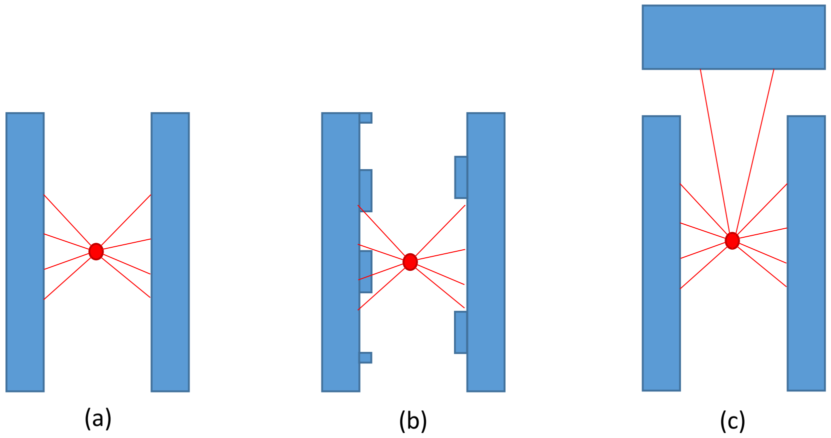


Figure 7-5 Special case that the vehicle self-localization longitudinal accuracy is low (a). In (b) the resolution of the map is good enough so we can see the longitudinal features and we can obtain the longitudinal position as well as lateral position. In (c) by adding a new wall in front, the vehicle can obtain the longitudinal position as well.

longitudinal position the accuracy degrades. The reason is that there is no observation difference between various positions in the longitudinal direction. This is true for almost any localization technique which uses laser scanner and map. In Figure 7-5(b), the resolution of the map is increased and more texture of the map appeared. In this situation, by using these features of the building, the longitudinal position can be obtained. Therefore, in this scenario, increase in resolution can raise the self-localization accuracy. In Figure 7-5(c), one more building added in the previous alignment of the buildings. In this scenario, the longitudinal position can be obtained using the front walls of the lateral building. In this case, there is no need for high resolution map because the layout of the map can cover the shortage in resolution.

As a conclusion, in some part of the map, plenty of building features are available which enables us to use lower resolution and therefore save the memory space. However, in some occasions few features are available and if the same resolution as the former is used, accuracy of the localization degrades by some extent. In this case, whole map can be made with higher resolution or just increase the resolution of some part of the map. therefore, some factors are necessary to evaluate the map in terms of accuracy.

7.1.2. Effect of resolution in NDT-based localization

In NDT-based map matching, instead of directly registering the Velodyne scan to the point cloud map, the scan is registered to the set of NDs called ND_{map} which is made

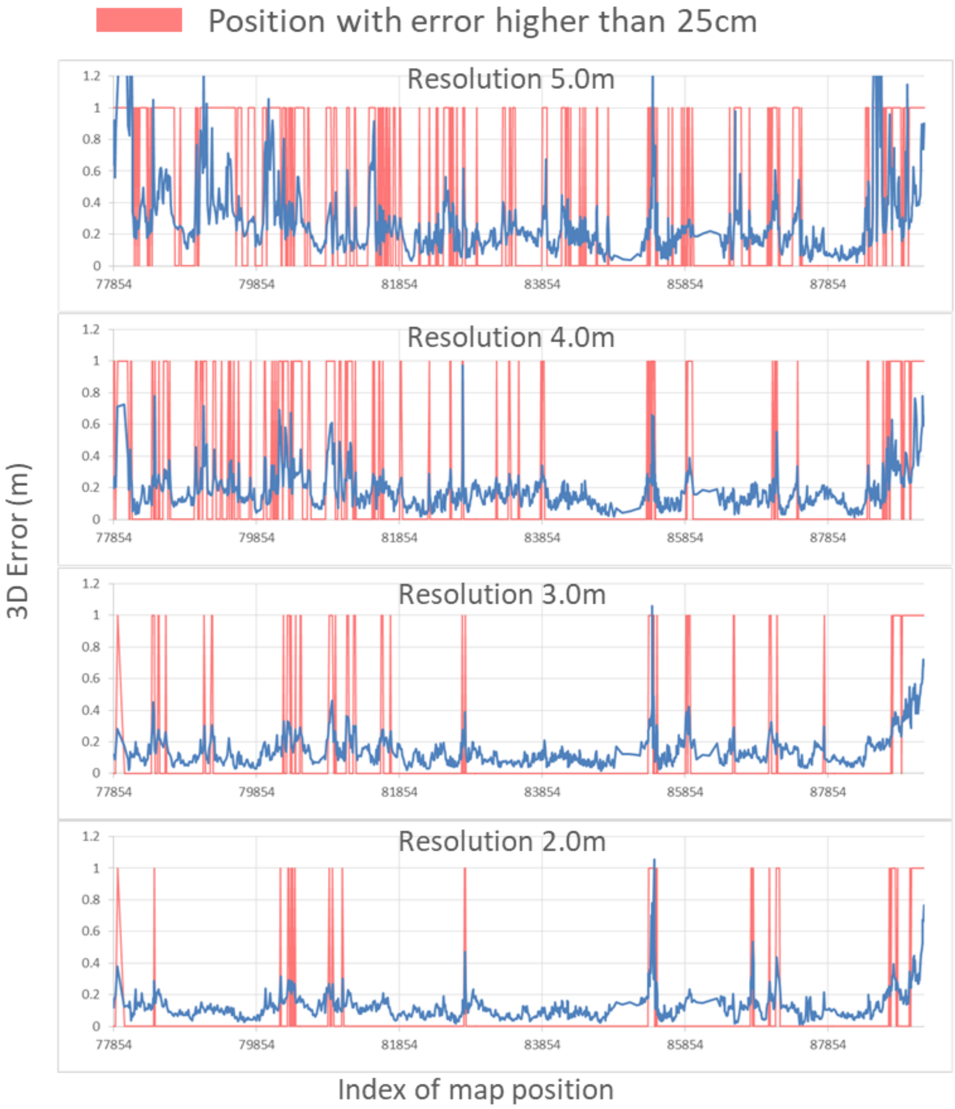


Figure 7-6 Localization error for different map resolution. Red region shows errors more than 0.25m.

from point cloud map. In order to make ND_{map}, map space is subdivided into fixed size voxels called cells. Then, from the points inside each cell, ND is obtained. To register the scan on this ND_{map}, the scan is moved over the NDs with 6D transformation matrix T and for each pose, likelihood are calculated. The scan pose correspond to the maximum likelihood is obtained by Newton's optimization algorithm and considered as the position of the vehicle within a map. For further discussion about NDT, readers are referred to [11], [70].

The error of self-localization within a map is highly related to the map resolution. Map resolution in the ND_{map} is related to grid size in the descritization process. If grid size of the map is low, it has higher resolution. Higher resolution means that the map can preserve more details of the environment. In other words, information loss in small grid size is lower. On the other hand, if the grid size of the map is big, details in the environemnt are ignored and there are more information loss. If the details of the map is ignored, localization error are more likely to appear. Figure 7-6 shows mean self-localization error of a specific path (Figure 7-7) for differet resolutions (2.0m ~ 5.0m). Mean error is obtained by averaging the localization error from different initial guess around each position. For each position, the initial guess is distributed in four sides with 0.2m interavals and range of 2m(total 441 initial guesses). Red region in Figure 7-6 shows localization error more than 0.25m. Figure 7-6 (5.0m) shows that in about 63.7% of the path, the error is less than 0.25m (white regions). This means if in those positions, ND_{map} with 5.0m resolution is used, still, high accuracy localization results can be obtained. On the other hand, in the red region which their error are more than 0.25m, the higher resolution should be used. In 4.0m resolution, 17.3% of the positoins has error higher than 0.25m and in 3.0m and 2.0m grid sizes, 10.5% and 7.1% respectively.

From Figure 7-6, it is clear that the error for specific grid size is different from place to place. One of the reason is that the amount of information loss is different from place to place as the volume of the details of the environment varies from place to place. Thus, in some places, more abstraction does not lead to more information loss. Moreover, in some places, more information loss necessarily does not result in localization error because of other characteristics of the map.

In fact, information loss is not the only reason for the localization error. Other criteria affect the localization error as well. One of these criteria is a sufficiency of features. If the features for matching is not sufficient or the quality of the features is not satisfactory, the error might happen. The layout of the features might be important as well. There might be places on the map that the number of high-quality features are enough. However, as all features faced parallel to the moving direction of the car, the longitudinal position of the vehicle cannot be obtained accurately, and localization error occurs. Tunnels, highways and deep urban canyons are some of the example scenarios that the layout of the environment cause localization error in the moving direction. On the other hand, in some places of the map, there might be similar features that make the localization

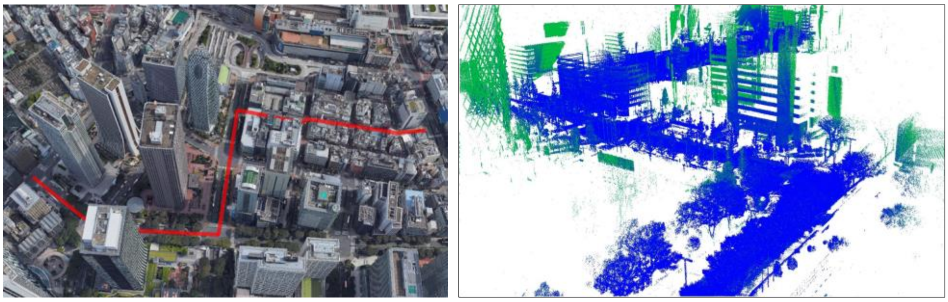


Figure 7-7 Path of experiments in Shinjuku, Tokyo (left), and its point cloud map (right).

erroneous, as the optimization process cannot match the input scan to the correct correspondences. Thus local similarity in the map can cause an error as well.

Therefore, in order to evaluate the ability of map in a specific position, these criteria should be considered together. Moreover, by evaluating these criteria, and feeding back it to the abstraction process, resolution of the map is adjusted to abstract the map as much as possible considering the required localization accuracy.

In order to clarify the discussion, in Figure 7-8, four scenarios are showed. In Figure 7-8, redpoints shows the input scan and bluwe ellipsoids are ND_map. The ellipsoids arround the vehicle shows the uncertainty in pose of vehicle. Red elliposoid means the error in localization. In Figure 7-8 (a), as the resolution is low, information loss happen, and other criteria cannot aid the localization as well. Thus the positioning has a high error in moving direction. However, in Figure 7-8 (b), one lateral feature in the top of the vehicle appears. In this situation, the vehicle can use these features for longitudinal positioning. Thus in this scenario, localization can be done with the same resolution without error. In fact, in (b), the layout of the features cover the information loss effect to the localization. In (c), the layout of the features and feature sufficiency cover the information loss effect on the localization. Information loss in (a), cannot be rectified by other factors because they are not satisfactory as well. Thus, in (a), by increasing the resolution (decrease the grid size), localization accuracy should be achieved (Figure 7-8 (d)).

7.1.3. Map refinement based on factors

In Chapter 5, map factros to evalaute map capability for self-localization is defined. In this section, these map factros are used to predict the error of each map resolution and define the best resolution for the map. Map factros are calcutaled and evaluated from low

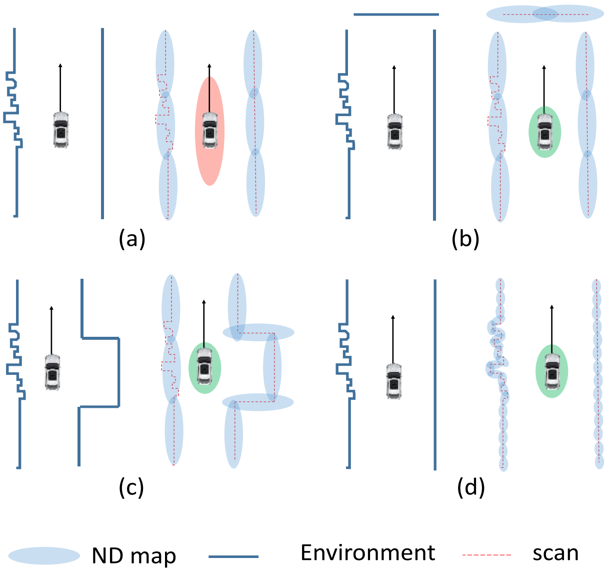


Figure 7-8 Vehicle position uncertainty in different scenarios. Red ellipsoid means high error and green ellipsoids means low position uncertainty.

resolution (big gridd size) to high resolution (low grid size) thus the process is called resolution refinement.

First sample point $P_{i,j}$ is distributed in map space with a 1.0m interval in x and y directions. These sample points are the points that the map evaluation are performed. Then for each sample point $P_{i,j}$, by calculating the factors and evaluating them, resolution of the local vicinity are obtained adaptively. To obtain resolution in the vicinity of point $P_{i,j}$, first, local vicinity of $P_{i,j}$ are extracted. Second, ND_{map} of local vicinity is generated with bigest grid size (here 5.0m). Third, using map evaluation factors defined in cahpter 5, the self-localization ability of this ND_{map} is evaluated. In this process, from the map factors, mean error are modeled (based on method in Chapter 6). In order to model the localization error based on the afformentioned factors, we use principal

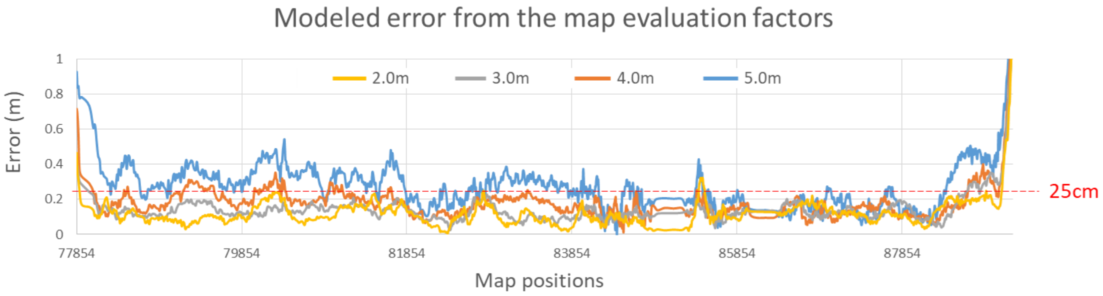


Figure 7-9 Modeled error using map evaluation factors for different grid sizes (2.0m~5.0m)

component regression (PCR). PCR is a regression analysis technique that is based on the principal component analysis (PCA). PCR applied to a set of factors as explanatory variables and actual mean localization error as a response. By doing this mean error for each position can be modeled from factors. Figure 7-9 shows the modeled error from map evaluation factors. These results are made utilizing the final map refinment procedure wchich uses the neural network for modeling the error.

If the modeled error from factors is more than error threshold, the resolution should be increased, and again the ND map is generated and evaluated. Error threshold in this work is set to 0.25m and 0.5m. This iterative refinement continues until the mean error becomes less than the threshold or the grid size reaches the lower value (here 1.0m). The result of this refinement for each position $P_{i,j}$ is the best resolution for the local vicinity of that position and stored in the $G_{i,j}$. G is a 2D arrray and has the same number as sample points $P_{i,j}$. This process is applied to all points in sample points in the map and 2D array G is filled.

Now, for each point $P_{i,j}$ in the map, the best resolution is stored in $G_{i,j}$. In order to make the ND_{map} from G, overlap region of each element should be considered. Suppose the value of $G_{m,n}$ is 1.0 and the value of its neighbor element $G_{m,n+1}$is 4.0. This means that for the local vicinity of the point $P_{m,n}$ the best grid size is 1.0m and for the $P_{m,n+1}$the best grid size is 4.0m. If the range of laser scanner is assumed to be 50.0m, local vicinity of these two points has 49m overlap. Therefore, for the overlap region, smaller grid size should be applied.

So for the final step, overlap region of the elements of G should be processed. For this, first, elements with the smallest value (1.0) is selected. Then, their value are overwritten into the elements in the range of laser scanner and their flag are set to one. Here, the flag is used to distinguish the elements which is already finalized. Next, for the remaining (flag=0), elements with second smallest value (2.0) are selsected and their vicinity with the range of laser scanner is overwritten with their value and flags are set to one. Overwrite is only applied to the elements which their flag is 0. This iterative process is applied to all grid size (1.0 ~ 5.0m). In this work the step of refinement is 1.0m. Figure 7-10 shows an example of before and after applying the final process to i_{th} row of G. In this example laser range is assumed to be 20m. As can be seen, small grid size

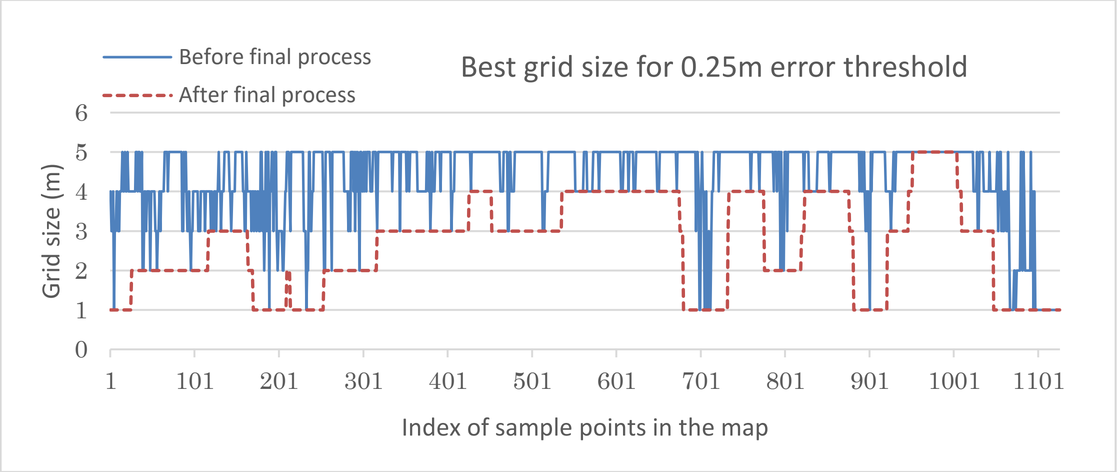


Figure 7-11 Best grid size of the map for experimental path before (blue) and after (red) final process for 0.25m error threshold.

in G force neighbor elements to be small too, thus overal abstraction ratio of map increase. Before applying final process, abstraction ratio of this row was 3.78 and after final process abstraction ratio became 1.99.

7.1.4. Experimental results

In order to evaluate the refinement strategy, experiments were conducted in Shinjuku, Tokyo. Experimental path and its point cloud map are shown in Figure 7-7. Length of this path is around 1.2Km. For the localization, VLP-16 is used. Frequency and laser range of VLP-16 is set to 20Hz and 25m respectively.

Map refinement strategy is done based on the map factors. Thus, first, we evaluate the effectiveness of the defined factors by modeling the mean error. By applying PCR to the factors of each grid size, the error is modeled based on factors.

Figure 7-11 shows the best resolution estimated based on the modeled error and 0.25m error threshold for the experimental path. Figure 7-12 shows same results for the error threshold of 0.5m. Blue graph is before final process and red is after final process. As Figure 7-11 shows, in most of the position in the path, 3.0m grid size can achieve required localization accuracy.

Experimental results show that our adaptive resolution strategy can shrink the map size of the experimental area down to 32.4 % compared to the 1.0m resolution while the average of the mean errors is less than 0.14m.

TABLE 7-1 Comparison of error for proposed adaptive map refinement strategy and different grid sizes.

	2.0m	3.0m	4.0m	5.0m	**Adaptive refinement (0.25m error threshold)**
Average of mean error (m)	0.140	0.154	0.193	0.283	**0.141**
Error more than 0.25m (%)	7.2%	10.6%	17.4%	36.3%	**9.9%**
Abstraction ratio compared to 1.0m grid ND map	3.57	7.31	11.98	16.76	**10.08**
Map size ratio (%) compared to 1.0m grid ND map	28.0%	13.6%	8.3%	5.9%	**9.9%**

TABLE 7-1 shows the average of the mean error for different grid sizes. If the 5.0m grid is used for entire area, an average of the mean error become 0.283m. Also, more than 36.3 percent of the cases errors more than 0.25m appears. Compared to this, by using the adaptive strategy, an average of the mean error for the experimental path is around 0.141m and only in 9.9% of the cases error more than 0.25m appears. This 9% caused because of error in the modeling phase. One of the solutions for overcoming this problem is to add more map factors in the phase of map evaluation. Another solution is to perform iterative refinement with a lower threshold than 0.25m.

Figure 7-13 shows best estimates for the map. Different colors of the positions show different resolution for their local vicinities. Red shows lowest resolution which is 5.0m and green shows highest resolution which is 1.0m. In (b, c) as many structured buildings

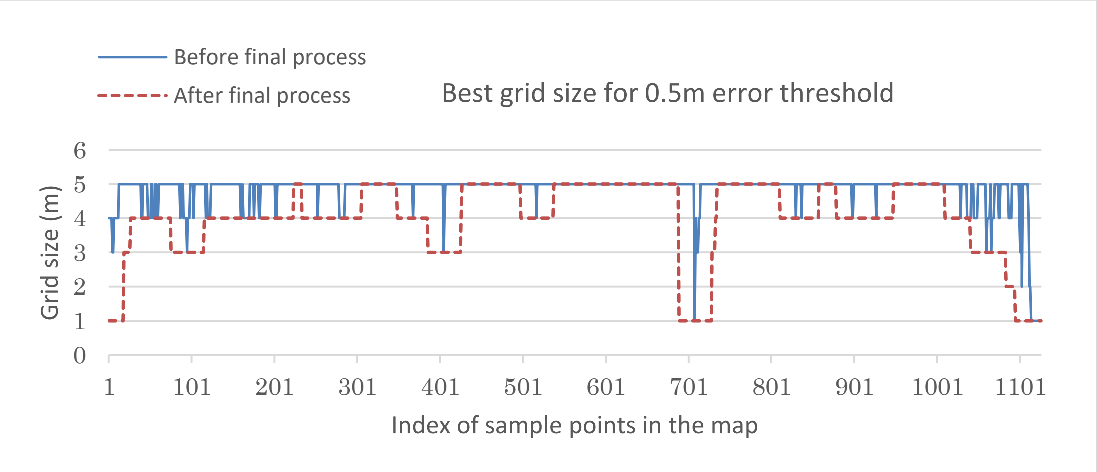


Figure 7-12 Best grid size of the map for experimental path before (blue) and after (red) final process.

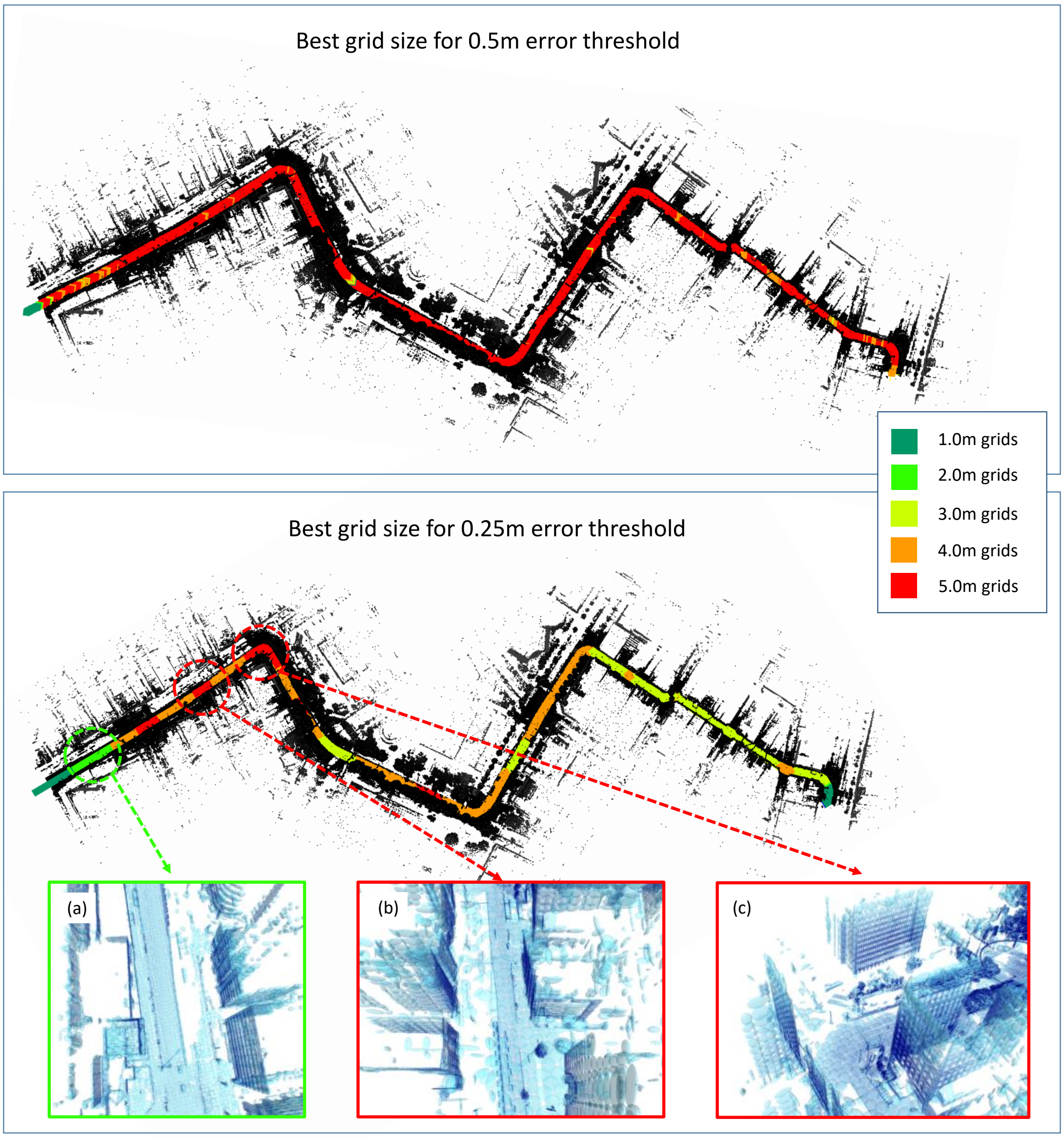

Best grid size for 0.5m error threshold
🟩 1.0m grids
🟩 2.0m grids
🟨 3.0m grids
🟧 4.0m grids
🟥 5.0m grids
Best grid size for 0.25m error threshold
(a)
(b)
(c)

Figure 7-13 Estimated best grid size for the experimental path. Top figure is for 0.5m error threshold and bottom figure is for 0.25m error threshold. Different colors shows different grid size. (a) 2.0m resolution. (b,c) 5.0m resolution.

surround the sample point, map factors have satisfying values, and thus the localization can be performed even with the low resolution of the map. Figure 7-13 (b, c) shows generated ND_{map} which has a 5.0m resolution. However, in (a) lateral features are missing. In other words, in this part of the map, perpendicular walls to the moving direction of the vehicle is few. Thus in this sample position, the localization error should be high. In this scenario, by increasing the resolution of the map, more details of the building is preserved in the map to help longitudinal positioning. Thus in this region, the grid size become 2.0m. Figure 7-13 (a) shows generated ND_{map} which has 2.0m resolution.

By use of these factors, mean localization error have been modeled with the goodness of fit of R^2 = 0.71 and $RMSE$=0.126. Based on this map evaluation, for each position of the map, best resolution have been estimated with map refinement strategy so that the map size shrink down to 9.9% while the mean error remain less than 0.141m.

7.2. Adaptive determination of the laser range

Accuracy and time efficiency are two key parameters of self-localization techniques for autonomous driving. While the scanning range of the 360 degree laser scanners have significant effect on the both accuracy and computation time of the map matching-based localization, this effect is not well investigated in the literature. In this section the effect of scanning range on the autonomous vehicle's localization is investigated. Based on this effect, a novel localization idea which defines the scanning range adaptively to improve the time efficiency while keeping the localization accuracy is proposed. To model the effect of scanning range on the localization accuracy in every local vicinity of the map, map factors from Chapter 5 are used. The capability of the proposed framework is verified using field data, demonstrating that it is able to improve the average matching time from 142.2 ms to 38.1 ms while keeping the localization accuracy within 10 cm.

7.2.1. Effect of laser range in self-localization

To evaluate the contribution of the laser scanning range on the map matching-based localization, an experiment was conducted near Shinjuku, a dense urban area of central Tokyo, Japan. Length of the experimental route was 1.4 Km and it included multiple crossings, narrow streets, wide streets, buildings, and trees.

The localization was performed using the laser scanner mounted on the roof and basic NDT map matching technique [62], [84] with different laser scanning ranges starting from 10 m to 50 m. For each scan sequence, the matching was performed using different initial guesses in a 1.0 m range of the ground truth position and the mean error and matching time were reported.

Figure 7-14 demonstrates mean localization error for the experimental route comparing to the laser range. By increasing the scanning range, the localization error

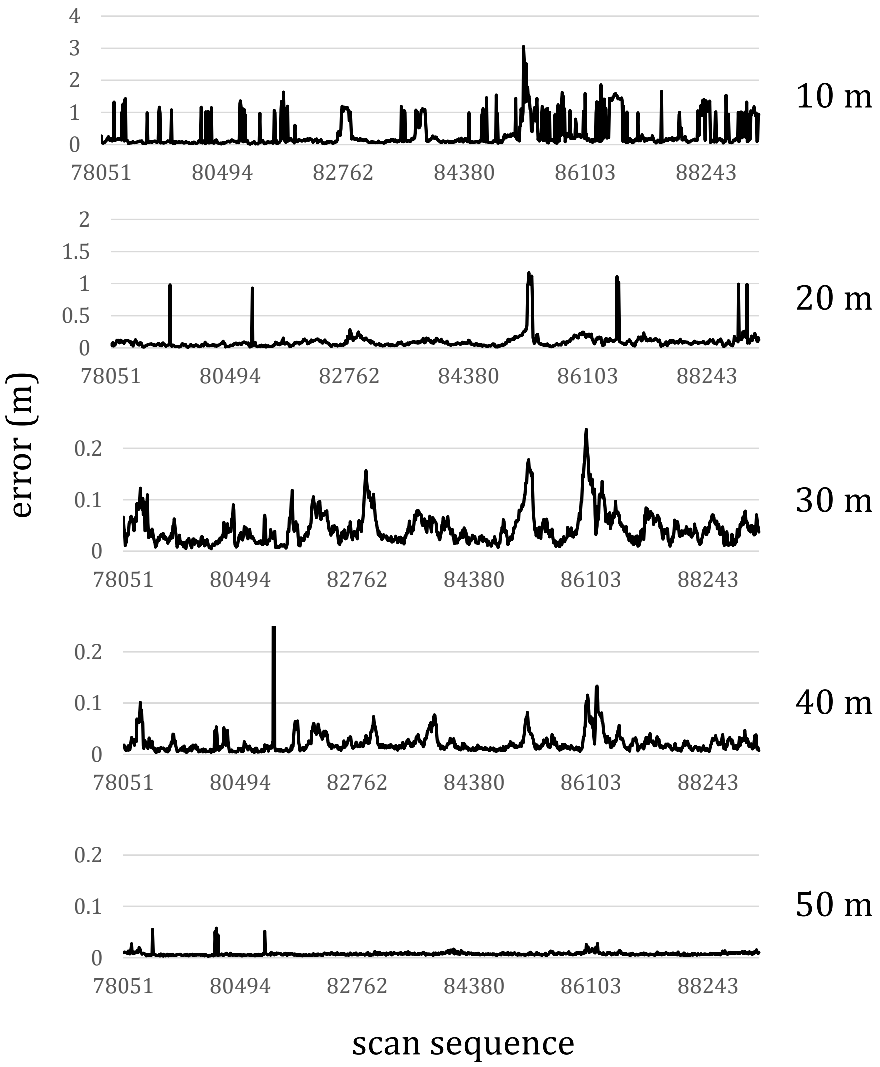


Figure 7-14 laser range vs localization error. As the laser range increases the localization error decreases.

decreases, and the 50 m scanning range has the best localization accuracy. Figure 7-15 shows the mean matching time for different scanning ranges. As shown in the figure, the matching time increases by increasing the laser scanning range for the localization. All the computations were conducted on an off-the-shelf PC with an 8-core, 3.50 GHz Intel Xeon E3-1270 V2 CPU and 16 GB of RAM running the 64-bit version of the Ubuntu 16.04 operating system.

The main reason that affects the matching time in Figure 7-15 is the number of points within the input scans. In the NDT matching, the input scan is first subsampled and then the matching is performed. In this work, the scan is down-sampled with 1.0m grid

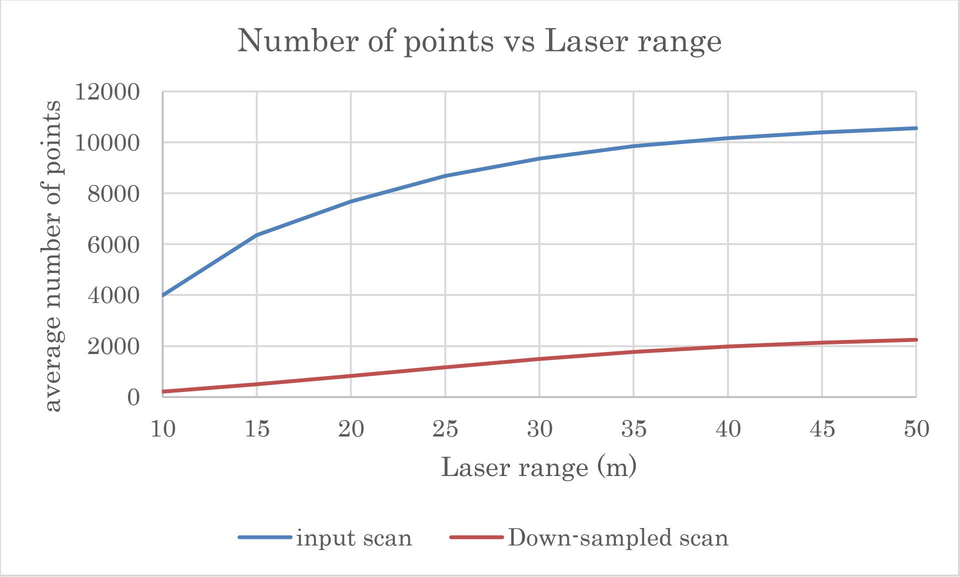


Figure 7-16 Average number of points in each input scan of laser scanner vs laser range in Path I. As the laser range increases the number of points also increased. Red graph is for the input scan after down-sampling with 1.0m grid size

size. Figure 7-16 shows the mean number of points in each scan sequence before and after

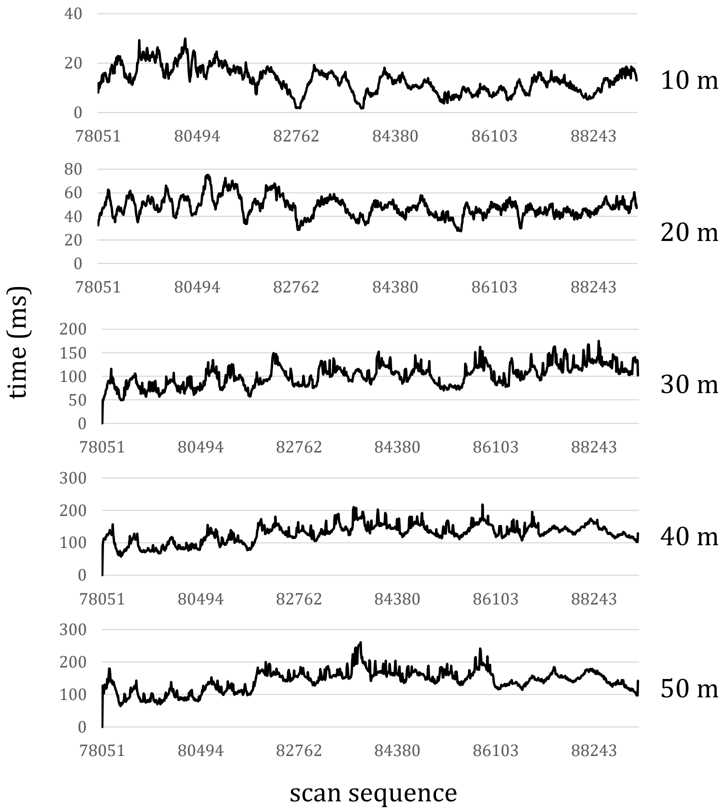


Figure 7-15 Matching time vs laser range. As the laser range increase the matching time also increases.

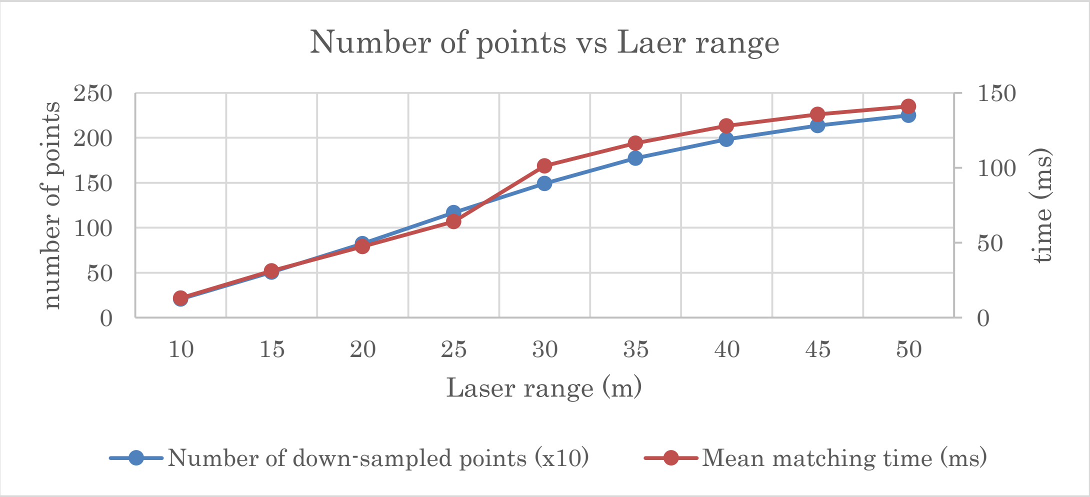


Figure 7-17 Comparison between the number of points and matching time.

the subsampling. By taking a look at Figure 7-15 and Figure 7-16 it can be understood that the matching time and number of points after the subsampling are in a direct and linear relationship, which means that by decreasing the scanning range time efficiency of the matching can be improved (Figure 7-17). Figure 7-18 shows the mean number of iterations in the NDT matching. As already mentioned, Newton's optimization algorithm iteratively investigates for the optimum answer. If the features are enough for matching it can be merged in final answer faster. In Figure 7-18, number of iteration decrease as the range of laser scanner increase. However this effect is covered by the increase of number

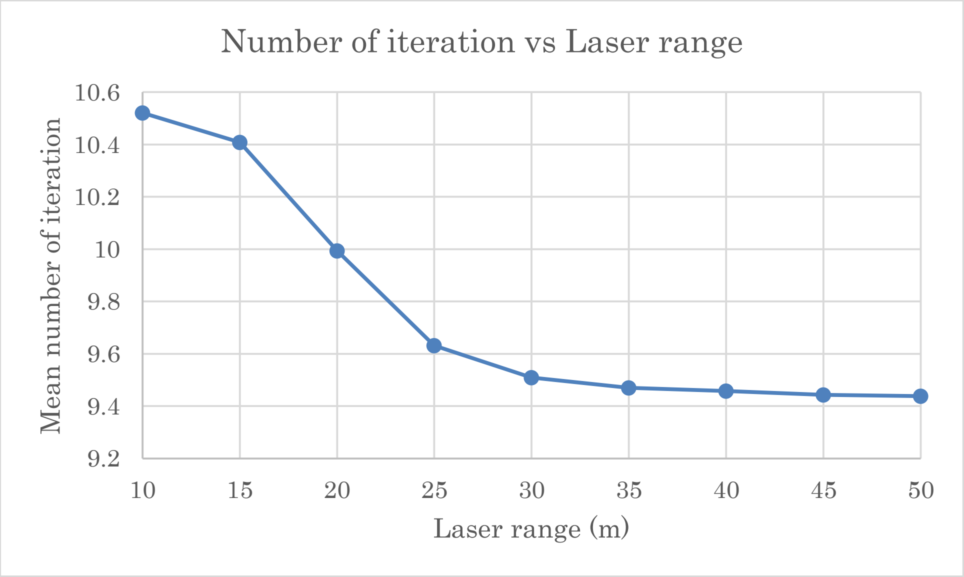


Figure 7-18 average number of points in each input scan of laser scanner vs laser range in Path I. as the laser range increase the number of points also increased.

of points and as a total, the matching time increase.

7.2.2. Adaptive determination of laser range

In this section, a dynamic scanning range localization method which aims to improve the time efficiency of the map matching while keeping the required localization accuracy is introduced.

As mentioned before, the scanning range affects both the accuracy and matching time. For the autonomous driving, a certain localization accuracy is expected (e.g. 25 cm) while a lower matching time preferred. It means that the scanning range should be increased only if the accuracy requirement is not met. In the ideal case, consider that the localization error is known beforehand for different scanning ranges starting from 10 to 50 m with 5 m intervals. In this case, the smallest range which meets the accuracy requirements can be chosen. Figure 7-19 shows the smallest scanning ranges during the experimental route that keep the localization error less than 10 cm and 25 cm.

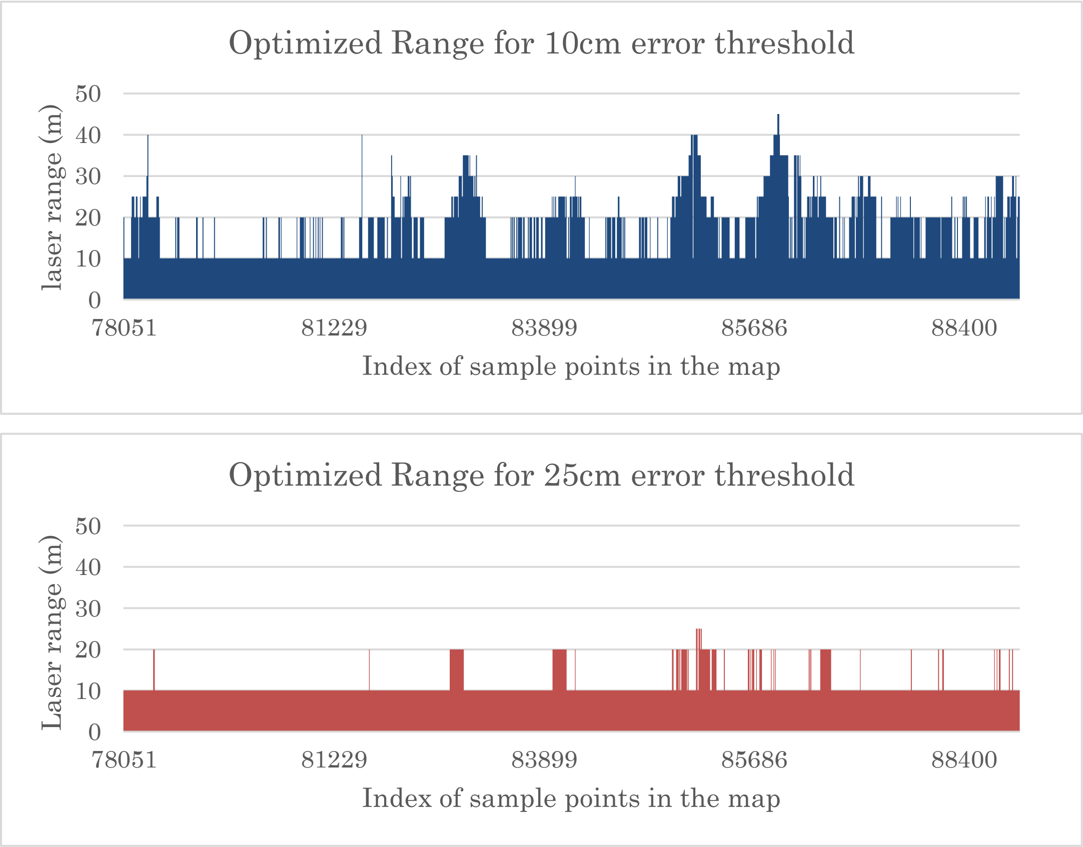


Figure 7-19 Optimized laser range for different sample points in the map for two error threshold. In top chart (red), laser range is selected so that the error become less than 10cm and in bottom chart (red), error threshold is 25cm.

TABLE 7-2 Matching time Evaluation

Matching time	**Scanning range**		
	50 m	**Dynamic range (10 cm error)**	**Dynamic range (25 cm error)**
μ (ms)	142.2	41.7	16.4
max (ms)	261.1	156.8	54.9

By applying the dynamic scanning range method, the average matching time for the experimental route reduces from 142.2 ms to 16.4 ms which is a significant improvement. This method not only improves the mean time, but also reduces the maximum matching time from 261.1 ms to 54.9 ms. TABLE 7-2 shows the mean and maximum matching time for the entire route using static scanning range of 50 m, dynamic scanning range for 10 cm error and 25 cm error.

To be able to define the dynamic range for each scan sequence in the previous section, it was considered that the localization errors for different scanning ranges are known beforehand. But it is not a realistic assumption and that is why defining the dynamic range which meets the accuracy requirement is not an easy work.

In this section, map evaluation factors introduced in chapter 5 are used to model the error and based on the error modeled, the laser range determination is performed adaptively. For this, map space is divided into 2D grids with 1.0m intervals. For center of each grid, the factors are calculated. Factors are calculated based on the range of laser scanner which is increased from minimum scanning range (10m) to maximum (50m) with 5m intervals. If the error modeled from factors become less than predefined error threshold (10cm, 25cm), the laser range is fixed to that value for that specific grid. Therefore for each grid, the optimum range of laser scanner are defined and this value is stored in the map to be used later in the localization phase. In the localization phase, based on the prediction of the next position of the vehicle (initial guess), optimum laser range is determined from 2D array which is already stored in the map.

7.2.3. Experimental results

In order to evaluate the effectiveness and accuracy of the proposed adaptive laser range determination framework, experiments have been conducted in the Shinjuku, Tokyo. The length of path for the experiments are 1.4Km. For this region, predefined laser range is calculated as described in the subchapter 7.2.2 and already stored in the map.

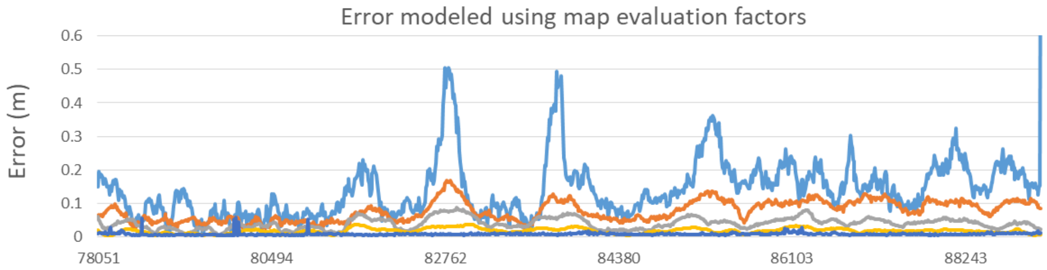


Figure 7-20 Error modeled from map evaluation factors.

Then by using this predefined values, adaptively the range is changed and localization are preformed. Modeled error using map evaluation factors are shown in Figure 7-20. Range of laser scanner for error threshold of 10cm and 25cm is shown in Figure 7-21. In most of the cases the localization can be performed with LIDAR range less than 25m. Blue graph is for the model that generated based on 10cm error threshold and red graph is the matching time which is based on the model generated with 25cm error threshold. As the error threshold in red is higher, more errors are acceptable, thus the range of scan can be smaller and thus the matching time is faster. However, due to miss modelling of the error of the map, in some cases localization error might extent from the threshold. Table X shows the error values for proposed method. TABLE II shows the error of localization using this adaptive range determination. In some places the prediction from factors are not accurate enough, thus the features in the defined rage is not enough and thus the

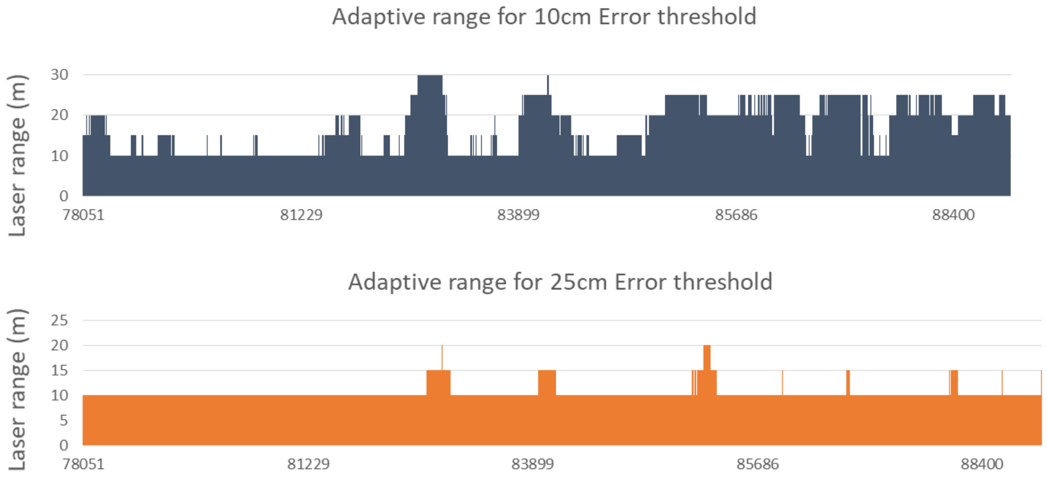


Figure 7-21 Adaptive range determined by comparing to map evaluation factors. (Top) for error threshold of 10cm. (Bottom) for 25cm error threshold.

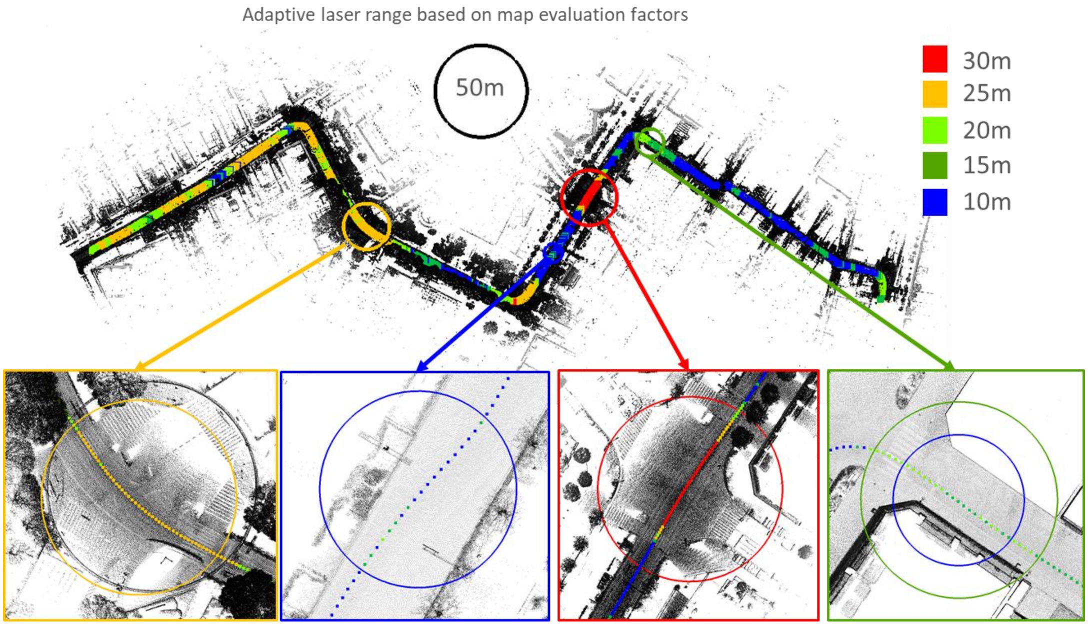


Figure 7-22 Adaptive range determined by map evaluation factors for 10cm error threshold.

localization cannot be performed with required threshold. The mean error for path I is 15.2 cm which is acceptable.

Chapter 8.
Conclusion

In this dissertation, we have introduced map evaluation factors to evaluate and quantify the map ability for self-localization. The focus has been made on the map based self-localization methods and several contributions have been made.

In the map based categories, map plays a significant role in achieving high accuracy self-localization. For accurate self-localization, the global and local accuracy of the map is essential as well, and many types of research have been done to obtain such a highly accurate map. However, the highly accurate map does not guarantee the accuracy of the localization. In other words, map accuracy is different than the ability of the map for localization.

Therefore, in this dissertation, four general criteria for the map has been defined. These criteria are feature sufficiency, layout, representation quality, and local similarity of the map. These criteria are defined regardless of the map format and can be applied to any other map formats. However, in this work, in order to quantify each of these criteria, the focus has been made on the ND map format, and several factors has been defined. For each point in the map, these factors are calculated based on the features in surroundings called local vicinity. By obtaining the correlation of the map factors with localization error, the effectiveness of the factors has been investigated. Additionally, by applying principal component regression (PCR), the predictability of the self-localization error based on these factors has been investigated. To evaluate the predictability of the defined factors, experiments have been conducted in Shinjuku, Tokyo, Japan. The route of experiments was around 40Km. The experimental results showed that the error modeled from the factors can represent the localization error of the map in 76.1% of the cases with an error lower than 10cm using PCR. In order to increase the accuracy of prediction, the factors have been fed to a simple feedforward neural network to model the error. The result has been improved and 78.2% of the localization error can be modeled with lower than 10cm accuracy.

As the other contribution of this dissertation, the outcome of the proposed map evaluation framework has been utilized to the adaptive refinement of the map and localization parameters. These parameters are map resolution and range of the laser scanner.

In addition to the aforementioned contributions, globally accurate urban mobile mapping frameworks have been proposed. One of the challenges of current mapping systems is their precision, accuracy and the cost. In the proposed urban mapping frameworks, two laser scanner has been used in addition to the Inertial measurement unit (IMU), odometer, and GPS. The airborne image has been used as a reference to correct the trajectory of the vehicle. In order to correct the trajectory of the mapping vehicle, the lane markings extracted from tilted laser scanner was registered to the lane marking extracted from the airborne image. The map which has been used for this dissertation was all made by this mapping system and global accuracy of the map is very high.

In this dissertation, two self-localization methods based on abstract map format which are multilayered vector map and the probabilistic planar surface map have been proposed as well. In the multilayered 2D vector map, different layers of building footprints are stored as a vector in the map. By using different layers of the building, the feature for matching increases and make the matching more accurate. For each vector, the variance of the points are stored in the map to allow the localization process to rely on more certain vectors rather than an uncertain one. This can reduce the map size to several thousand times. In the probabilistic planar surface map, instead of using heavy point cloud, the planar surfaces are extracted from the point cloud map. Each planar surface contains the variance of the points which shows the uncertainty of the planes. Storing the variance for each plane help the localization process to rely more on the certain walls. In the matching phase, distortion of the input scan is one of the challenges which makes the misalignment of the input scan to the map. In this work, distortion is removed in the optimization process. In fact, in each iteration of the optimization, the scan is reshaped by the new results and the distortion is removed to fit better to the map.

One of the possible future works for map evaluation framework is to use deep neural network. In this case, the local vicinity of the sample point is directly fed to the network and the task of defining factors are all given to the network

Appendix

GCL Social Innovation Project

Towards Autonomous Vehicle Platform for Smart Society

Overview

More than a century ago, the first production vehicle was made by Karl Benz and later automobiles dramatically changed the way we live. Today, the autonomous driving technology has a potential to revolutionize not only the transportation, but also the entire society. Every year, more than a million lives of people are cut short due to traffic accidents while the autonomous vehicles can significantly reduce these fatalities. This emerging technology can also save huge amounts of time for the people in the society. If we start working when we get into the car, how many hours we can save? Especially in metropolitans like Tokyo! In addition, autonomous driving facilitates transportation of aged people, reduces the fuel consumption and plays a key role in saving the environment, reducing the parking spaces, and so on.

To achieve these opportunities, an intensive cooperation between academia, industries, government and society is required to boost the research, make a reliable business model, review and modify the laws, and prepare the society for the required changes (Figure A. 1.). Even within the academia, an interdisciplinary collaborations between the researchers in the field of the autonomous driving and the other fields such as city planning, sociology, psychology, and environmental science is required.

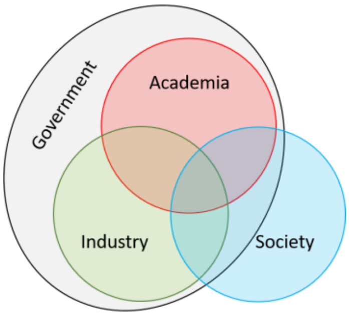


Figure A. 1. Cooperation required for benefiting from the autonomous driving technology

Social Innovation Project

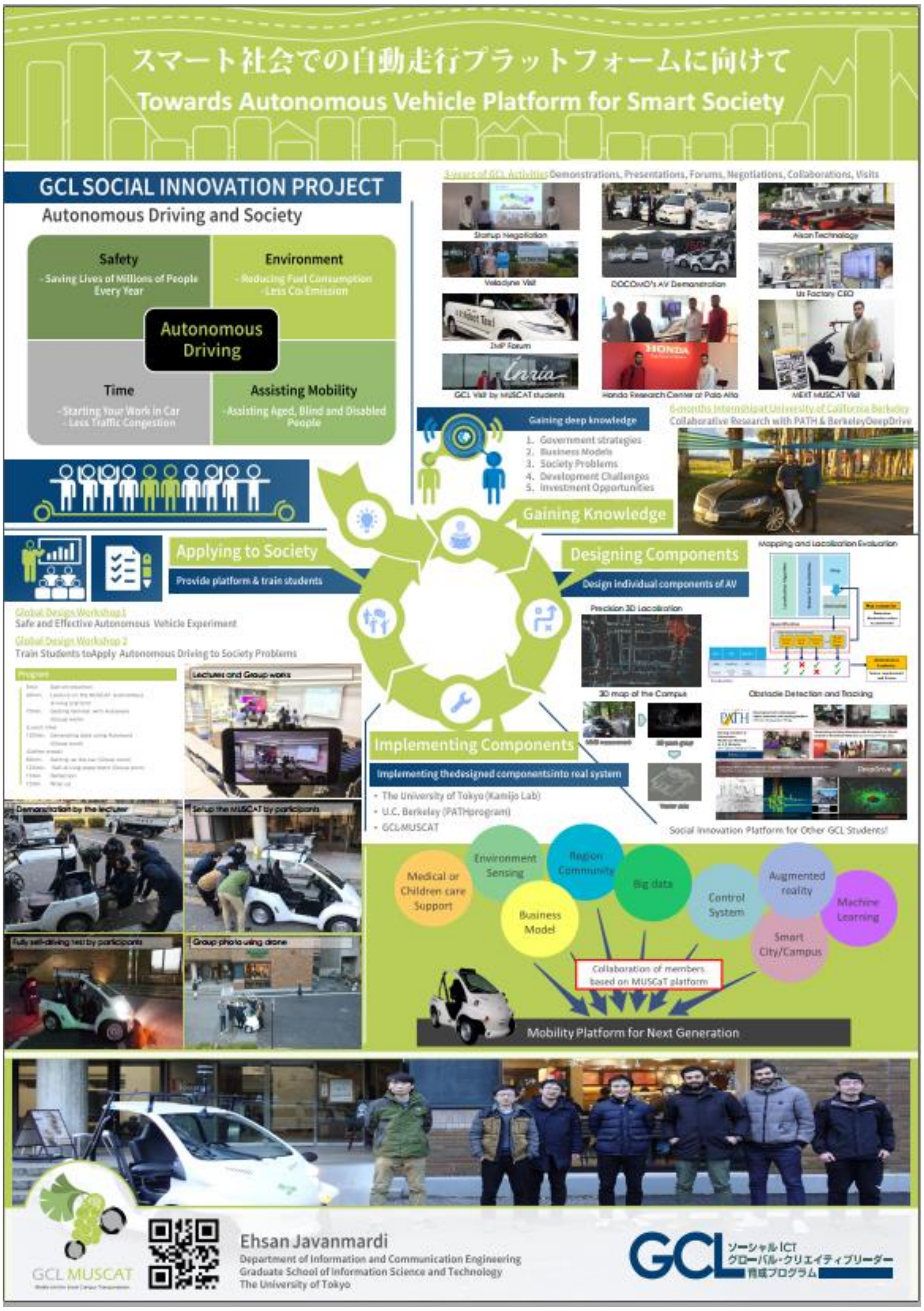
The aim of this social innovation project is to provide the GCL students with an autonomous driving platform while initiating a small scale cooperation between academia, industries and society. To achieve this goal, four phases of 1) Gaining the knowledge and initiating the cooperation; 2) Design the components; 3) Implement the components; and 4) Apply to society; are designed and followed.

The members of this project is Ehsan Javanmardi and Mahdi Javanmardi who worked together during their Ph.D. program from the early stages to the end. This project is a collaboration with the GCL-MUSCAT (Mobile Unit for Smart Campus Transportation) under the supervision of Dr. Tsukada and all outputs will be accessible for the GCL students to boost their own social innovation project. Overview of author's GCL project is shown in Figure A. 2. Each phase of the project and achievements are described in the following.

1) Gaining the knowledge and initiating the cooperation

In order to design a platform which is going to make a revolution in the society, only a high-end technology cannot promise the success. Besides, the knowledge about the government strategies, business models, society problems, development challenges, and investment opportunities are required. To gain these knowledge and widen the networks, the author contributed to several projects, demonstrations, meetings and conferences with industries and academia. Some of these contributions are listed and described below.

- **Networking and Collaboration with Industries and Government:**
    - **ITS-Japan:** ITS-Japan was launched in January 1994 as Japan's response to the developments intelligent transportation systems, and it enjoys the full backing of National Police Agency, Ministry of Economy, Trade and Industry, Ministry of Land, Infrastructure and Transport, National Police Agency, Ministry of Public Management, Home Affairs, Posts and Telecommunications, and Ministry of Construction. The author joined ITS-Japan's sub-working group meetings as a representative of the Kamijo Lab, the University of Tokyo and learned the high-level management issues of implementing the autonomous driving in the society from the frontiers in the



スマート社会での自動走行プラットフォームに向けて

Towards Autonomous Vehicle Platform for Smart Society

GCL SOCIAL INNOVATION PROJECT

Autonomous Driving and Society

Safety
- Saving Lives of Millions of People Every Year

Environment
- Reducing Fuel Consumption
-Less Co2 Emission

Autonomous Driving

Time
-Starting Your Work in Car
- Less Traffic Congestion

Assisting Mobility
-Assisting Aged, Blind and Disabled People

3-years of GCL Activities: Demonstrations, Presentations, Forums, Negotiations, Collaborations, Visits

Gaining deep knowledge

1. Government strategies
2. Business Models
3. Society Problems
4. Development Challenges
5. Investment Opportunities

Gaining Knowledge

6-months Internship at University of California Berkeley
Collaborative Research with PATH & BerkeleyDeepDrive

Designing Components

Design individual components of AV

Precision 3D Localization
3D map of the Campus
Mapping and Localization Evaluation
Obstacle Detection and Tracking
PATH
DeepDrive

Implementing Components

Implementing thedesigned componentsinto real system

- The University of Tokyo (Kamijo Lab)
- U.C. Berkeley (PATHprogram)
- GCLMUSCAT

Social Innovation Platform for Other GCL Students!

Applying to Society

Provide platform & train students

Global Design Workshop 1
Safe and Effective Autonomous Vehicle Experiment

Global Design Workshop 2
Train Students toApply Autonomous Driving to Society Problems

Program

Lectures and Group works
Demonstration by the lecturer
Setup the MUSCAT by participants
Fully self-driving test by participants
Group photo using drone

Collaboration of members based on MUSCaT platform

Mobility Platform for Next Generation

Ehsan Javanmardi
Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

GCL ソーシャルICT グローバル・クリエイティブリーダー 育成プログラム

Figure A. 2. Overview of the social innovation project "Towards Autonomous Vehicle Platform for Smart Society".

field. The accuracy requirements for the localization and mapping of an autonomous driving platform, statics about the mapping whole Japan for autonomous driving, and state-of-the-art of the mapping technology is a few examples of the knowledge gained during these meetings and discussions.

- **Honda-HRI:** The author had collaborated with Honda Research Institute (HRI), USA during two projects with Berkeley Deep Drive (BDD), "Generic Motion Generation and Comprehension with Social Interactions[1]" and "3D Object Detection based on Lidar and Camera Fusion[2]". The main contribution of the author was generating a high-precision map from raw data collected by HRI. The resulting map was used for the motion generation and object detection of autonomous vehicle. During this collaboration, the author learned how a frontier car industry like Honda invest to the academia to keep their technology always updated. Also the author's research was qualified for the shortlist in the Honda Y-E-S Forum 2018.
- **ATEC Co. and Aisan-Technology:** Map generation is one of the most important basics for the autonomous driving. Therefore, to design an autonomous vehicle platform, a deep knowledge about the state-of-the-art of the mapping technology and its current challenges is required. To gain this knowledge, the author had several meetings and discussions with ATEC Co. and Aisan-Technology, two frontier mapping companies of Japan, to know the details about their aerial mapping systems and mobile mapping system (MMS). Even the author joined their data collection experiment in a challenging environment such as Hitotsubashi intersection around Tokyo station with a lot of tall buildings. This knowledge helped the author in the design and implementation phase of the autonomous vehicle platform.
- **Valeo Japan:** Valeo is a multinational automotive supplier based in France, providing a wide range of products to auto manufacturers and after-markets. The author had opportunity to negotiate with Valeo Japan, the Japanese arm of the Valeo Group, about possible collaborations and their investment to initiate a startup for the high-precision mapping and localization of the autonomous vehicles. Also, by invitation from their CTO, the author applied

[1] https://deepdrive.berkeley.edu/project/generic-motion-generation-and-comprehension-social-interactions

[2] https://deepdrive.berkeley.edu/project/3d-object-detection-based-lidar-and-camera-fusion

to the “Valeo Innovation Challenge 2018[3]”.

- **Networking and Collaboration with Academia:**
  - **GCL MUSCAT:** In the GCL MUSCAT project, with a collaboration of Dr. Tsukada, the author could initiate an autonomous driving hub in the University of Tokyo which every student with a different background can join, learn the basics about the autonomous vehicles, contribute to the project, and benefit from the autonomous driving platform in his/her research. In fact, GCL MUSCAT was a hub for inter-academia collaboration and networking for autonomous driving. As a core member of GCL MUSCAT, the author could connect several international project leaders to cooperate in the future, including the leaders UC Berkeley Deep Drive, TierIV, Islamic Azad University.
  - **U.C. Berkeley:** To gain the required knowledge, the author widen his networks internationally with frontiers in the field of autonomous vehicle’s perception, motion planning, and control. During collaboration with PATH and BDD at U.C. Berkeley the author could gain valuable knowledges about how to promote a research, academia-industry collaboration, working with members with different backgrounds.
  - **Kamijo Lab:** The Kamijo lab was playing a central role in this academia networking and collaboration. Other than above mentioned parties, many other valuable connections were made through Kamijo lab, including research groups from Saitama Institute of Technology, Nagoya University, Riken-AIP, etc.
- **UC Berkeley internship:** To experience the state-of-the-art research in the international groups and collaborate with the industries, the author had a 6 months internship at the Institute of Transportation studies of University of California at Berkeley (UCB). During this visit which has been done from 2016 Sep to 2017 Feb, the author collaborated with Partners for Advanced Transportation Technology (PATH) under the supervision of Prof. Wei-Bin Zhang. During this visit, many

[3] https://valeoinnovationchallenge.valeo.com/en/challenges/innovationchallenge2018

research institute and companies have been visited in order to gain the knowledge and experience. Some of these companies and research groups are, Velodyne Co.[4], which is the biggest producer of the-state-of-the-art multi-channel laser scanners, Honda research institute[5] in Palo Alto, Berkeley deep drive[6] (BDD), etc. During this visit the author has contributed to several projects which are listed below:

i. Build a multi-modal data collection platform for gathering synchronized data from multiple sensors. These sensors are Velodyne HDL-64 which has 64 channel data, 4 HD cameras which can see surrounding of the car, IMU, GPS, RTK-GPS, and odometer data from CAN. Figure A. 3 shows the Lincoln car which is equipped with these sensors and the data logger software.

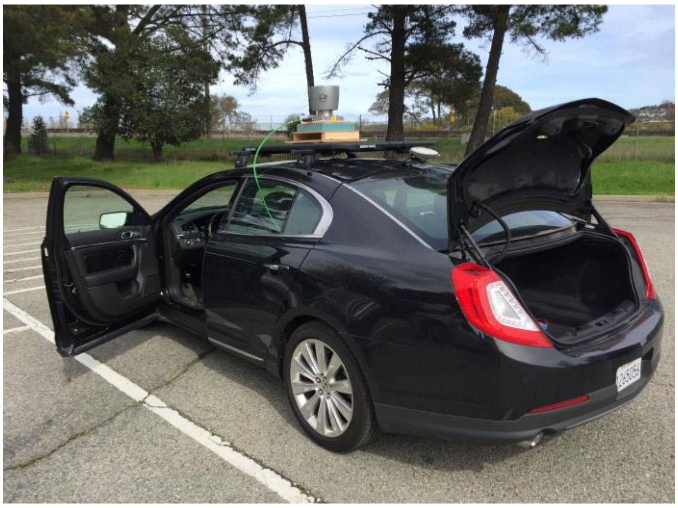


Figure A. 3. The designed data collection platform for autonomous driving

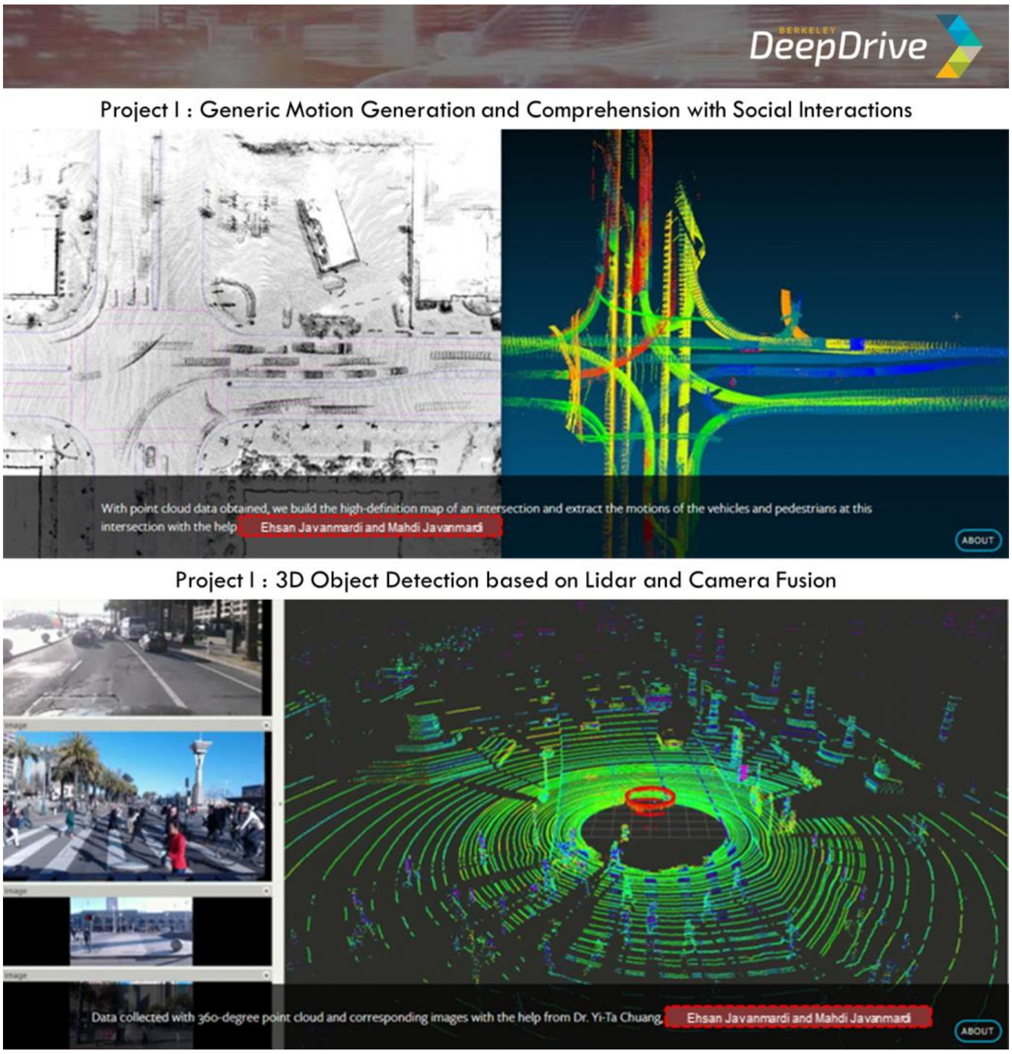
ii. Vector map of several locations of Berkeley which was generated with a collaboration of the Honda Research institute and BDD. The Figure A. 4 shows the website of the BDD which is show that the author are collaborated in some of the projects.

iii. Obstacle detection algorithm with implementation of the clustering algorithm which uses the angle of the points not distance of the points.

[4] www.velodynelidar.com/

[5] https://usa.honda-ri.com/

[6] https://deepdrive.berkeley.edu/



Project I : Generic Motion Generation and Comprehension with Social Interactions

With point cloud data obtained, we build the high-definition map of an intersection and extract the motions of the vehicles and pedestrians at this intersection with the help Ehsan Javanmardi and Mahdi Javanmardi

Project I : 3D Object Detection based on Lidar and Camera Fusion

Data collected with 360-degree point cloud and corresponding images with the help from Dr. Yi-Ta Chuang, Ehsan Javanmardi and Mahdi Javanmardi

Figure A. 4. The contribute projects on the website of the BDD

iv. ITSS ITS Now offers a quarterly, free, limited-time access to articles from the IEEE ITS Society publications and conference proceedings. Each issue has a dedicated editorial team of topical experts. Each team selects 10-15 outstanding papers from Society publications IEEE Explore access is granted to the full text of these ITS papers. It is launched in 2013 and three numbers are published so far. The fourth topic was selected as self-localization. The author was responsible for writing the surveys about the papers in the IEEE ITS society and related to self-localization. 40 papers are selected and the survey was written. However the ITS Now letter is not yet published.

Through these valuable experiences we could get familiar with Government strategies, companies business models, current society problems, development challenges and also consider the investment opportunities. This knowledge equipped the author with a set of tools to design and implement a comprehensive autonomous vehicle platform. The generated network in this phase is illustrated in Figure A.5.

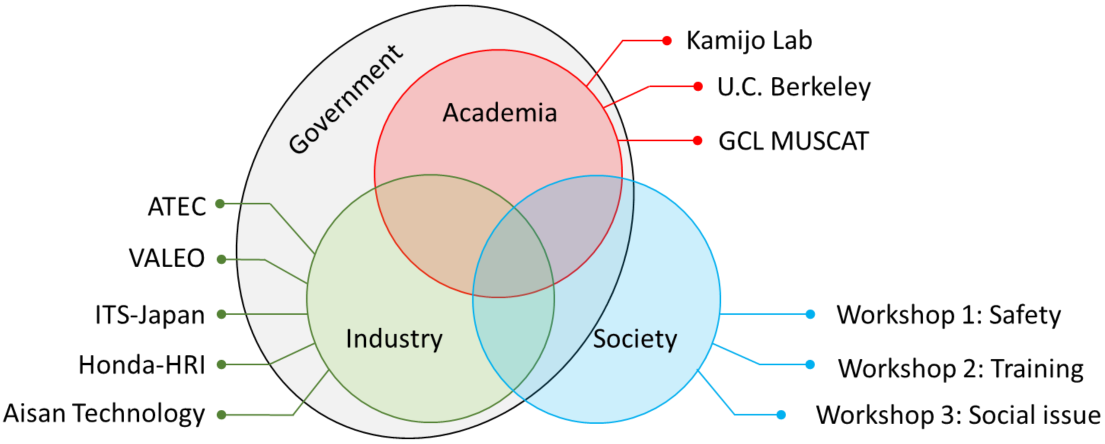


Figure A. 5. Overview of the generated network in the knowledge phase

2) Designing the components

For the autonomous platform for smart society, the author with collaboration of the other team member (Mahdi) designed three main components of the autonomous driving, which are mapping, localization, and obstacle detection. In the following, the role of this author in designing each component are described.

- **Mapping and Localization:**

Mapping and localization are two key requirements for autonomous vehicles and these two topics are the main topics of the Ph.D. dissertation of this author. Currently most of the AV platforms use HD map which is a point cloud map of the path and laser scanner to get the position of the vehicle. This point cloud map is very huge in size and it is very difficult to store and to be downloaded. Thus in part of this author Ph.D., abstracted map formats which are multilayer 2D vector map and probabilistic planar surface map is proposed to be used instead of point cloud map. The details of these two algorithms are mentioned in Chapter 4 of this dissertation. By using these map formats, the size of map can be shrink to more than 1000 times and yet the accuracy is preserved. These two localization components and map formats are designed and later implemented and applied in the MUSCAT platform. To make the code compatible with the Autoware software which is used in MUSCAT, these modules are implemented using Robot Operating System (ROS). All design and implementations of these localization systems are done by this author.

The other requirement of the autonomous vehicles are HD map. Currently, the HD map of Hongo campus is provided by AISAN TECHNOLOGY CO., LTD. However the map of campus is not enough for testing the platform for different scenario and applications. Thus the system which can map the environment for different places other than Hongo campus is essential. This system which is low-cost urban mapping framework is designed by this author and the details of this implementation is explained in Chapter 3 of this dissertation. By using designed low-cost urban mapping framework, some part of the Shinjuku are mapped and is ready for the use of MUSCAT platform. In addition to this, the proposed mapping framework can be used for mapping any other places which is needed. In order to design the mapping framework, the idea of automatic calibration of the MMS data to the airborne imagery is used which is proposed by other MUSCAT team member (Mahdi). Also, some part of the data acquisition system which is used in this mapping framework is designed during the internship in the U.C. Berkeley and designed together by this author and other team member (Mahdi).

- **Obstacle detection:**

Obstacle detection is one of the important component for autonomous driving. The vehicle needs to be aware of the surrounding environment and detect the obstacles. This can be done based on various sensors such as vision, radar and LiDAR. In the designed component, Velodyne’s HDL-64 which has 64 channels is used. Design of obstacle detection program have been done by this author and Mahdi as a part of internship at the California PATH and this author have been contributed from the early stage to the end.

These designed components were presented in flagship conferences and journals in the field of intelligent transportation systems and intelligent vehicles. The localization component which was presented in the ITSC’17 was able to receive the best student paper award of the conference. The details of each component can be found in the publications listed at the end of this dissertation.

3) Implementing components

Each of the mapping, localization and obstacle detection component was implemented, tested, and evaluated in different projects during three years. For example, the localization component was fully implemented and tested in the Kamijo laboratory's experimental vehicle, mapping component was also implemented in the Kamijo laboratory with a collaboration of ATEC Co., and obstacle detection component was implemented in U.C. Berkeley during the internship period. All the components are designed using C++ and ROS in order to be compatible with MUSCAT platforms main program which is Autoware. These components which were initially designed and implemented separately were finally integrated into the MUSCAT platform and the outcome of these systems can be used by GCL MUSCAT autonomous driving hub.

4) Apply to the society

The final phase of the project is applying the platform to boost GCL student's social innovation project and achieve the opportunities provided by autonomous driving. This phase can be achieved through three Global Design Workshop C (GDWC). The 1st GDWC was organized on August 23rd, 2017 to help understanding the safety requirements for experiment inside the campus. This author was the co-organizer of this workshop. During the first workshop, GCL students experienced the autonomous driving by sitting in the fully automated vehicle and traversing inside the campus. The outcome of the workshop was the safety guidelines for working and doing the experiment with MUSCAT platform. The second GDWC was held on December 21st, 2017. In this workshop, which was the advanced version of the first workshop and this author was the organizer, the guidelines for setting up the platform are introduced and the GCL students are trained with the platform so that they can use the MUSCAT platform for their research without the help of the MUSCAT members. As one of the key component of the autonomous driving is mapping and localization, first a short lecture was given about mapping and localization. Then, the attendees were divided into two groups, and they setup the system and collect the required data and run the system according to the checklists and guidelines. By getting the feedbacks from GCL students we could modify the checklist and guidelines. The outcome of this workshop was the checklist and guidelines for using the platform and serval students who were trained to operate the platform. The third workshop will be organized in September 2018 which aims to define one social application of the autonomous driving and apply it using GCL MUSCAT platform.

The following will be the report of the GCL workshop #1 and #2.

GCL-MUSCAT Workshop #1

"Safe and Effective Autonomous Vehicle Experiments"

Safety first, autonomous driving next!

This workshop aimed to introduce the MUSCAT platform to the students who are interested in self-driving cars and their social benefits, and strengthening their skills in designing a safe and effective autonomous driving experiment. The workshop attendees were also asked to help the MUSCAT members to draft a safety manual by joining a brainstorm at the end of the workshop. In this workshop, this author was the co-organizer. Figure A.6, A.7 and A.8 shows some view of this workshops.

By attending this workshop, attendees could become familiar with the key components of the driverless cars, observe the step-by-step procedure of an autonomous

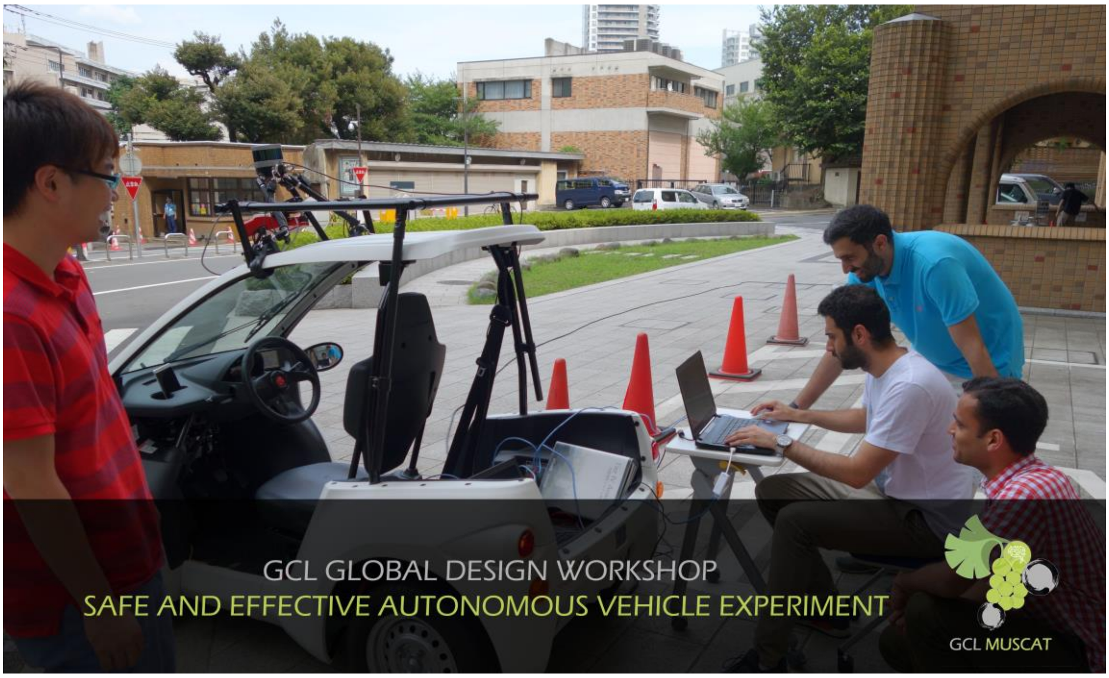

GCL GLOBAL DESIGN WORKSHOP
SAFE AND EFFECTIVE AUTONOMOUS VEHICLE EXPERIMENT
GCL MUSCAT

Figure A. 6. The announcement of GCL MUSCAT Workshop #1

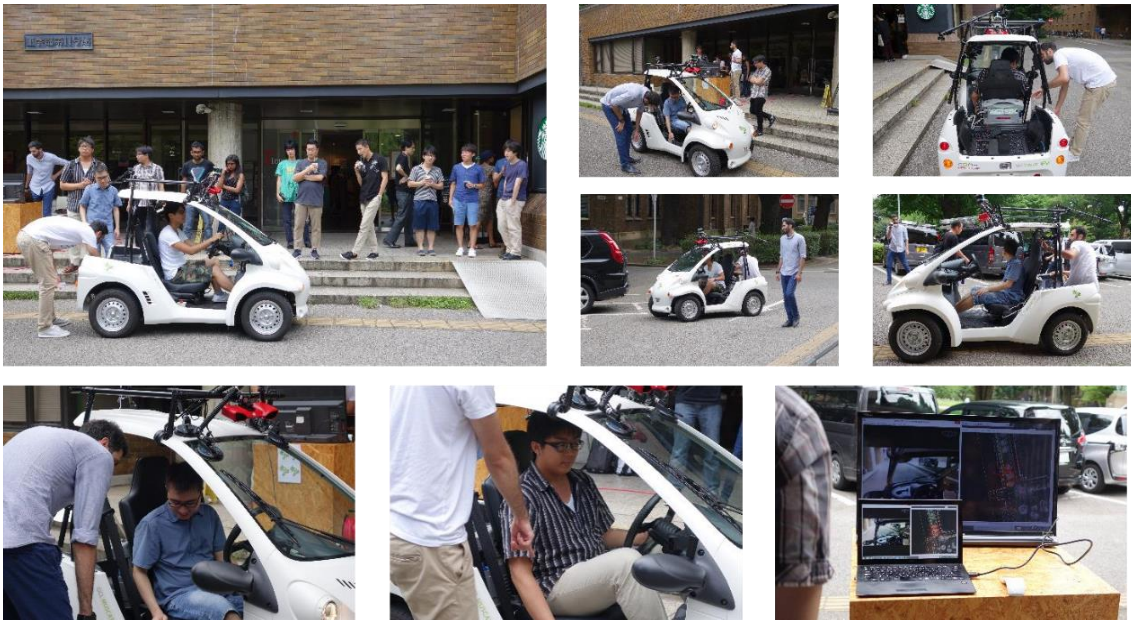


Figure A. 8. A view of the workshop 1 self-driving experience by students

driving experiment, and experience self-driving cars for themselves. During the self-driving experience, they had to pay attention to the requirements of safe experiment design consisting of experiment planning, required checklists, safety considerations, dealing with emergency situations, and how to report after the experiment. Moreover, this workshop gave the students an opportunity to think about how MUSCAT can assist their

Figure A. 7. Group photo of the GCL MUSCAT Workshop #1 research and social projects, and provide us with feedback to define our future development priorities.

The specific outcome of this workshop was a manual titled “GCL MUSCAT Safety Manual” which is currently being used by everyone who uses the MUSCAT platform. This workshop was an excellent opening for the MUSCAT global design workshop series, which will be followed by our next workshop titled GCL-MUSCAT Workshop #2 "Workshop on Autonomous Driving." The participants of this workshop are as follows: Information and Communication Engineering (3 persons), Institute of Industrial Science (2 persons), Institute of Artificial Intelligence and Robotics (1 person).

GCL MUSCAT Workshop #2
“Workshop on Autonomous Driving”

From basics to experience, let’s become an expert on self-driving cars!

More than a century ago, the first production vehicle was made by Karl Benz, and later automobiles dramatically changed the way we live. Today, fully autonomous driving has the potential to revolutionize not only transportation but also the entire society. Every year, more than a million lives are cut short due to traffic accidents—autonomous driving could significantly reduce these fatalities. This emerging technology can also save people time. If we start working when we get into the car, how many hours can we save? Especially in a metropolis like Tokyo. Also, autonomous driving facilitates the transportation of aged people, reduces fuel consumption and plays a key role in saving the environment, reducing the number of required parking spaces, and so on.

All the opportunities mentioned above require an interdisciplinary collaboration between autonomous driving experts and researchers in other fields such as city planning, sociology, psychology, environmental science, etc. The autonomous driving platform “GCL-MUSCAT” aims to provide GCL students with an opportunity to utilize self-driving cars and boost their social innovation projects to make our society a better place to live.

In our previous workshop “GCL-MUSCAT GDWS C #1, Safe and Effective Autonomous Vehicle Experiments” we demonstrated our autonomous driving platform to the interested GCL students and provided them an opportunity to experience autonomous driving on campus. The outcome of the first workshop was a safety checklist for MUSCAT that ensures our autonomous driving experiments meet basic standards of safety.

In this workshop, we aimed to extend our MUSCAT professional network and train more students to be able to operate this platform from scratch. Later, these students will be able to help and assist other GCL students from different fields to employ MUSCAT in their social innovation projects. To achieve this goal, the attendees were divided into two groups and received multiple sessions of in situ training and group work from 10 AM to 6:30 PM. The workshop consisted of four main components including a lecture to transfer advanced knowledge of autonomous driving to the attendees, indoor group work to familiarize them with the autonomous driving software Autoware and generate the required data, outdoor group work to train them on how to set up and operate the MUSCAT platform and do the experiment, and finally reflection and group discussion to get their feedback and increase their motivation for future collaborations.

【Items】

Autonomous driving vehicle, Autonomous driving software, PowerPoint, Whiteboard, Laptop, Drone, LCD, Wireless internet

【Participants】

Information and Communication Engineering (2 persons), Emerging design and informatics course (1 person), Institute of Artificial Intelligence and Robotics (1 person)

【About the WS】

The MUSCAT project is an interdisciplinary project in GCL supervised by Dr. Tsukada which aims to provide autonomous driving technology to GCL students who believe that this technology might boost their social innovation projects.

Since autonomous driving technology is new and complex, it is difficult for students with no prior knowledge of self-driving cars and programming to operate such systems and perform experiments by themselves. Therefore, an interdisciplinary collaboration between students who are majoring in information or computer science and students in other fields is essential to achieve the goals of the MUSCAT project.

This workshop focused on training new students and equipping them with the knowledge to operate the MUSCAT platform and organize autonomous driving experiments. This workshop aims to expand the professional network of MUSCAT members, but also to ensure knowledge transfer from current members to prospective members of MUSCAT.

This workshop started at the GCL-lab with a short self-introduction and continued with a 40-minute lecture on the MUSCAT autonomous driving platform. In the lecture, the main components and technologies of self-driving cars were described using examples. After that, the students were divided into two groups, and one laptop with Autoware (open-source autonomous driving software) was given to each group. With instruction from the facilitators, they started getting familiar with Autoware and manipulating the software by themselves.

To provide an environment that allowed the students to share their opinions with the facilitators, facilitators and students had their lunch at the cafeteria together. After lunch, members came back to the GCL lab and started the second round of group work to generate the required data for an autonomous driving experiment. When the data was ready, members moved the car outdoors, beside the Starbucks café on Hongo campus, and started setting up the car. After they prepared the system and data, each group performed the autonomous driving experiment, and every student sat behind the wheel while the car was in the autonomous driving mode.

After everyone tried the system, all the attendees and facilitators had a group discussion and reflection to get the students’ feedback on how to improve the experiment quality and define the development priorities of MUSCAT for the future.

Finally, the attendees and facilitators took the car back to the GCL lab, and the workshop was finished with a wrap-up. The video report of the workshop can be found here: https://photos.app.goo.gl/c5BjL8SMHw7i7oty1

【Program】

10:00-10:05	Self-introduction
10:05-10:45	Lecture on the MUSCAT autonomous driving platform

10:45-12:00	Getting familiar with Autoware (Group work)
12:00-12:45	Lunch break@cafeteria
12:45-14:45	Generating data using Autoware (Group work)
14:45-15:00	Coffee break
15:00-16:00	Setting up the car (Group work)
16:00-18:00	Self-driving experiment (Group work)
18:00-18:15	Reflection
18:15-18:30	Wrap up

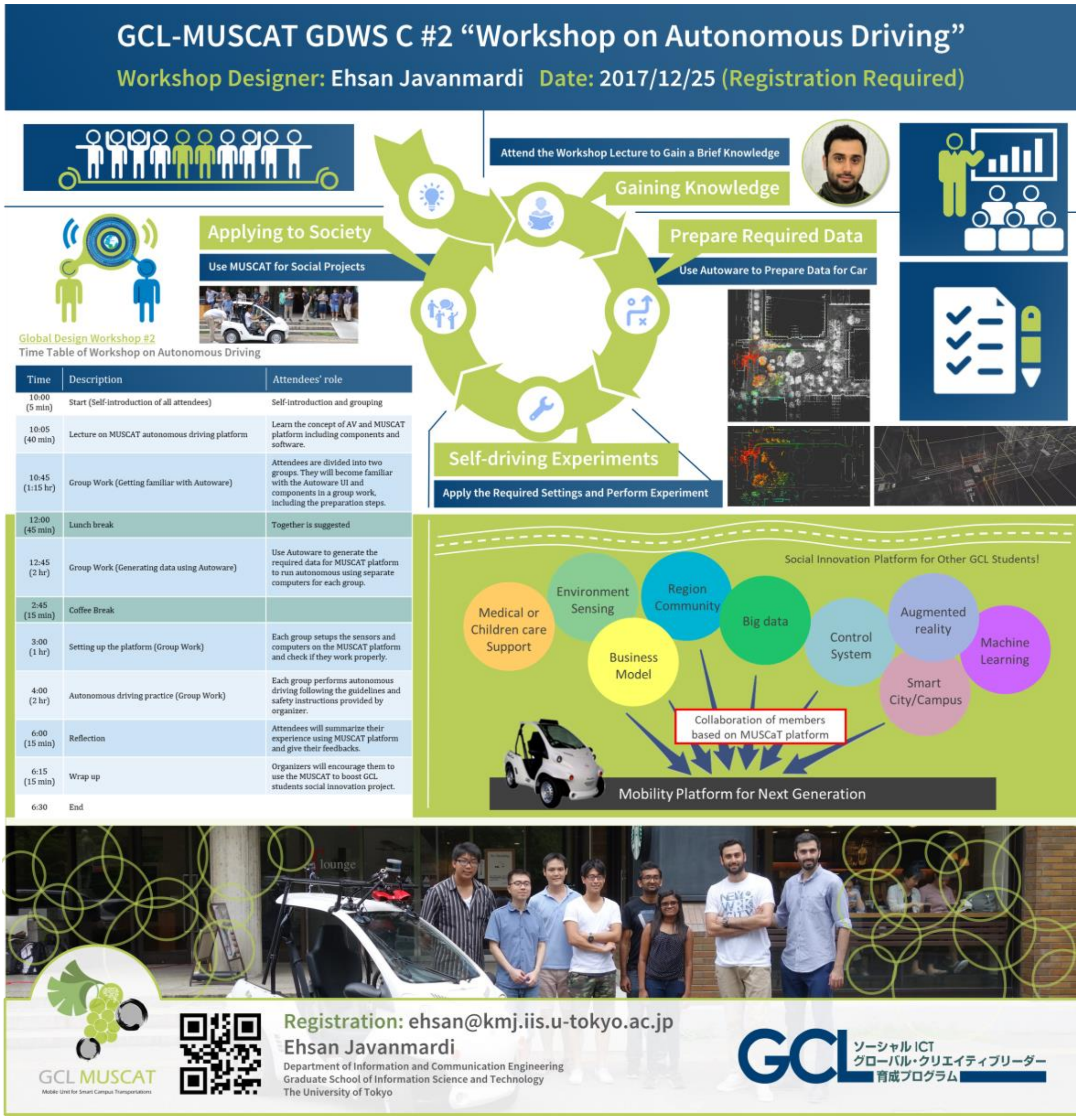
【Achievement】

The main achievement of this workshop was undoubtedly the new students who can operate the MUSCAT platform and will be able to organize an autonomous driving experiment from scratch without help from the MUSCAT core members, after just a few practices. Since current core members will graduate soon, these new students can fill the gap and assist the GCL students who are interested in the platform but cannot operate it by themselves. Moreover, similar workshops should be regularly organized to expand this network.

Another outcome of the workshop is a step-by-step experiment manual which will make the experiment procedure less complicated and easier to perform. This manual is currently under construction, and after its release, it will be modified from time to time to reflect each new experience and challenges members have faced during the experiments.

【Reflection】

The main goal of this workshop was to expand the MUSCAT professional network. One of the essentials for growing a project is to make more people interested in what you are doing. While autonomous driving is an attractive topic, many students are afraid of its complexity. Therefore, they are not easily convinced to join this project. To avoid this problem, we should promote MUSCAT in a way that every GCL student can feel they can join and benefit from this project without having any prior knowledge of autonomous driving or programming. To be able to achieve the goal of attracting people from different majors and backgrounds, this workshop or similar workshops are essential. The role of this kind of workshop is to strengthen the MUSCAT core members, and make a base for



GCL-MUSCAT GDWS C #2 "Workshop on Autonomous Driving"

Workshop Designer: Ehsan Javanmardi Date: 2017/12/25 (Registration Required)

Attend the Workshop Lecture to Gain a Brief Knowledge

Gaining Knowledge

Prepare Required Data

Use Autoware to Prepare Data for Car

Self-driving Experiments

Apply the Required Settings and Perform Experiment

Applying to Society

Use MUSCAT for Social Projects

Global Design Workshop #2

Time Table of Workshop on Autonomous Driving

Time	Description	Attendees' role
10:00 (5 min)	Start (Self-introduction of all attendees)	Self-introduction and grouping
10:05 (40 min)	Lecture on MUSCAT autonomous driving platform	Learn the concept of AV and MUSCAT platform including components and software.
10:45 (1:15 hr)	Group Work (Getting familiar with Autoware)	Attendees are divided into two groups. They will become familiar with the Autoware UI and components in a group work, including the preparation steps.
12:00 (45 min)	Lunch break	Together is suggested
12:45 (2 hr)	Group Work (Generating data using Autoware)	Use Autoware to generate the required data for MUSCAT platform to run autonomous using separate computers for each group.
2:45 (15 min)	Coffee Break	
3:00 (1 hr)	Setting up the platform (Group Work)	Each group setups the sensors and computers on the MUSCAT platform and check if they work properly.
4:00 (2 hr)	Autonomous driving practice (Group Work)	Each group performs autonomous driving following the guidelines and safety instructions provided by organizer.
6:00 (15 min)	Reflection	Attendees will summarize their experience using MUSCAT platform and give their feedbacks.
6:15 (15 min)	Wrap up	Organizers will encourage them to use the MUSCAT to boost GCL students social innovation project.
6:30	End	

Social Innovation Platform for Other GCL Students!

Medical or Children care Support

Environment Sensing

Region Community

Business Model

Big data

Control System

Augmented reality

Machine Learning

Smart City/Campus

Collaboration of members based on MUSCaT platform

Mobility Platform for Next Generation

GCL MUSCAT

Mobile Unit for Smart Campus Transportations

Registration: ehsan@kmj.iis.u-tokyo.ac.jp

Ehsan Javanmardi

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

GCL ソーシャルICT グローバル・クリエイティブリーダー育成プログラム

Figure A. 9. Poster of the second GCL workshops.

other students.

Beyond this, we should focus on one social problem and design GCL-MUSCAT workshop #3, which will target a specific social concern. If workshop #3 is held successfully, we will be able to extend this model to other social problems and involve more GCL students over time.

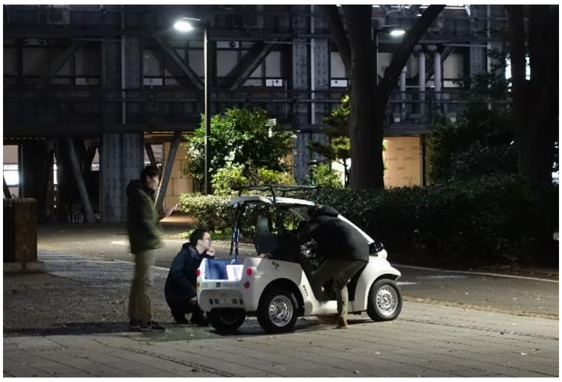


Figure A. 10. During the fully autonomous driving test, the attendees faced a problem several times and they fixed it.

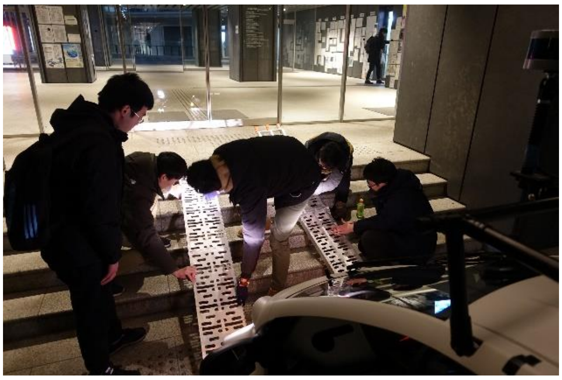


Figure A. 11. The MUSCAT platform were returned to the GCL-lab and workshop were finished.

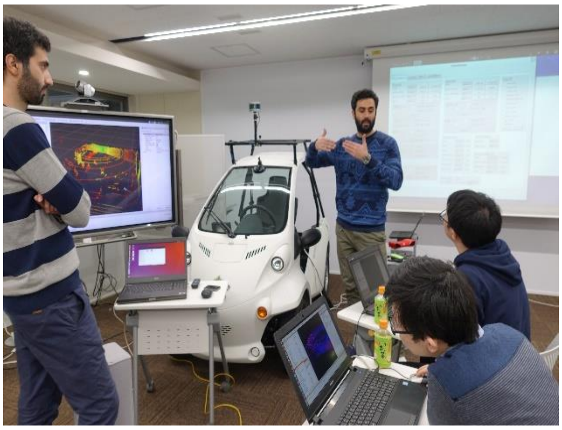


Figure A. 12. Lecture on autonomous vehicle, MUSCAT platform and Autoware by Ehsan Javanmardi.

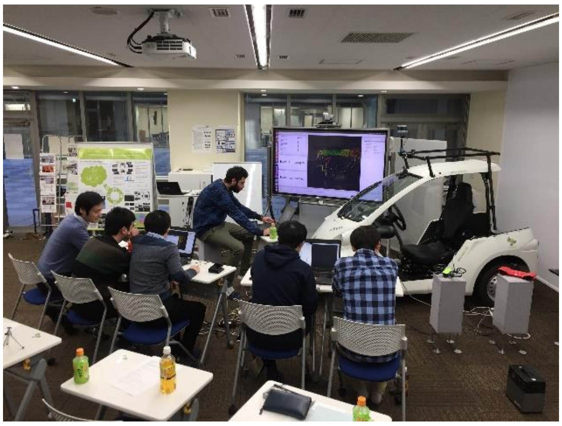


Figure A. 13. The participants were divided into two groups and they followed the steps explained by the lecturer.

Figure A. 14. After the lecture, the attendees run the software by themselves.

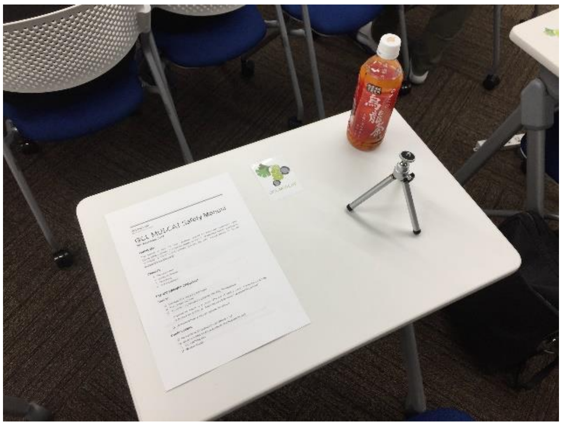


Figure A. 15. The safety manual as well as sticker of the MUSCAT logo were distributed among all participants.

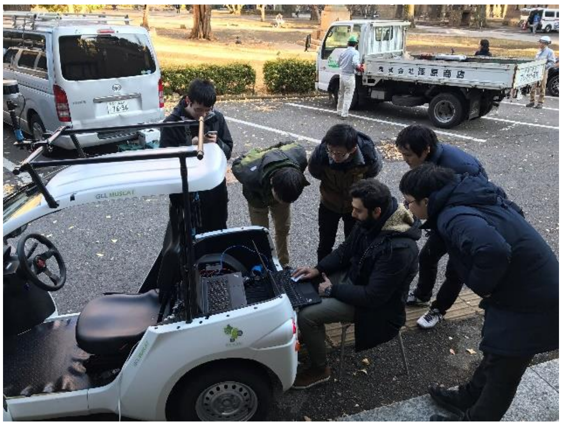


Figure A. 16. Instructors showed how the system should be set up.

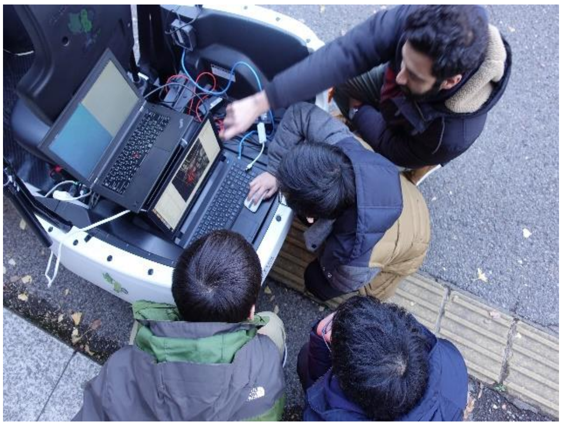


Figure A. 17. Participants tried to set up the system by themselves and apply what they have learned.

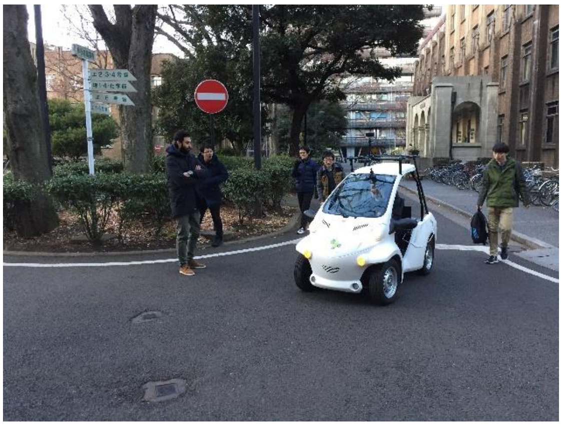


Figure A. 18. The participants experienced a partial autonomous driving which was set up by themselves.

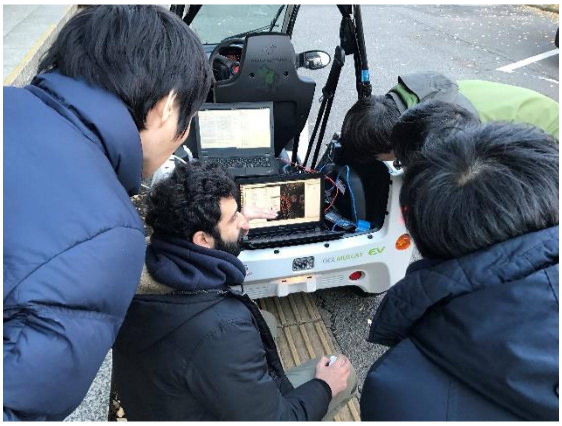


Figure A. 19. The system was set up to run on the fully autonomous mode.

Figure A. 20. Everything was double-checked before fully autonomous driving test.

Figure A. 21. Group photo was taken before the sunset.

Figure A. 22. Group photo was taken from the bird’s view of drone.

Figure A. 23. Fully self-driving test after the sunset.

Figure A. 24. Attendees exchanged their thoughts about today’s experience and difficulties they faced during the setup of the system.

Finally the outcome of the GCL MUSCAT Workshop #1, which is a safety manual is shown in the following. This author was the co-organizer of GCL MUSCAT Workshop #1.

GCL MUSCAT Safety Manual (Output of Workshop)

OVERVIEW

This manual provides the basic checklists required for performing experiments using GCL-MUSCAT’s robocar. This manual should be read by all attendees before organizing an experiment. If there is any question regarding the safety manual contact Dr. Tsukada (tsukada@hongo.wide.ad.jp).

CHECKLISTS

1. Pre-experiment
2. During experiment
3. Emergency
4. Post-experiment

PRE-EXPERIMENT CHECKLIST

Basics:

- □ This manual is read by all attendees
- □ This manual is printed
- □ At least two experienced experts are attending the experiment (Experienced expert is a person who has at least 3 times of autonomous driving experience using MUSCAT. Beginners should be trained by experts beforehand.)
- □ Understand how to manually operate the robocar

Confirmations:

- □ Permission of Dr. Tsukada (When, Where, Who)
- □ University program check (Festivals, Entrance exams, etc.)
- □ GCL Lab Programs
- □ Weather Check

Setup:

- □ Setup warning signs along the test route (Cones, warnings)
- □ Setup autonomous driving sign on the car (if available)

Emergency preparation:

- □ Understand how to stop robocar in case of emergency
- □ Write down the address of campus and experiments area for an emergency case
- □ Bring emergency kits

DURING EXPERIMENT CHECKLIST

Basics:

- □ Warning people (a person in front of the car for warning pedestrians)
- □ Driver's right foot on the break without pushing (the break overwrites the program mode)
- □ Driver's right hand on the handbrake (handbrake is physical and does not overwrite program)
- □ Front display's emergency stop button is not for emergency case

Pre-autopilot checks:

- □ Check laptop's batteries before starting autopilot (Autoware and Controller pc)
- □ Check if the electricity cable is plugged-out
- □ Driver is behind the wheel
- □ Fasten the seatbelt
- □ Driver's hand should not be inside the steering wheel
- □ Warning sound using speaker while robocar is on autopilot (if available)
- □ Handbrake should be pulled up before starting autopilot
- □ Confirm with every member before starting autopilot

EMERGENCY CHECKLIST

Basics:

- □ Keep calm!
- □ Emergency contacts:

Manabu Tsukada: (Mobile) 080-9558-2094, (email) tsukada@hongo.wide.ad.jp
GCL Office: (Tel) 03-5841-8746(内線 28746), (email) gcl.jimukyoku@gcl.i.u-tokyo.ac.jp

Driver:

- □ Push the emergency stop button
- □ Pull the handbrake
- □ Park the car in a safe place

Other members:

- □ Check the victim's health condition (If there is any victim)
- □ If the situation is severe call the **119** and ask for an ambulance
- □ Stay beside the victim till ambulance arrives

Reporting:

- □ Call Dr. Tsukada and GCL office
- □ Take picture of the situation, victim, car
- □ Write down accident report including following information and sign the report
  - □ Time and place
  - □ Contact information of the victim (phone number, plate number of car)
  - □ The reason of accident
  - □ Describe the situation (car speed, accident scenario, awareness of the victim, ...)
  - □ Attach the pictures
  - □ Describe the level of damage with pictures
  - □ Screen captures of the experiment (if available)
- □ Organize a meeting with responsible persons and GCL to avoid similar mistakes

POST-EXPERIMENT CHECKLIST

Basics:

- □ Park the car in GCL-Lab
- □ Put the key to the locker
- □ Plug the car into charger
- □ Plug the laptops into the charger
- □ Emergency or dangerous situation report (If any)
- □ Organize GCL-lab's chairs and tables
- □ Backup the data if required
- □ Share the pictures of experiments using Google Drive
- □ Make a quick report and share it using Google Doc including checklist photos

Conclusion

In this social innovation project, with Collaboration of GCL MUSCAT project which is started under supervision of Prof. Tsukada, an autonomous driving platform was provided for the GCL students while a small scale cooperation between academia, industries and society was initiated. To achieve the goals, four phases of 1) Gaining the knowledge and initiating the cooperation; 2) Design the components; 3) Implement the components; and 4) Apply to society; were followed. This author have had several contributions to the GCL MUSCAT project which are specially in the field of vehicle self-localization and HD map required for autonomous driving and the technological details have been mentioned in Ch. 3,4,5, and 6 of this dissertation.

Currently, all outputs of this project is accessible for the GCL students to boost their social innovation project by using autonomous driving technology through GCL MUSCAT platform and network hub.

GCL Acknowledgment

This research was supported by Graduate Program for Social ICT Global Creative Leaders (GCL) of The University of Tokyo by Japan society for the promotion of science (JSPS).

I would like to express my appreciation to Prof. Kamijo for giving me a unique chance to pursue the GCL program. I also want to thank Prof. Kuniyoshi, Prof. Asami, Prof. Naemura, Prof. Tsukada, Prof. Mizukoshi, Prof. Ohnishi, Prof. Aida, Prof. Gharebaghi, and all other GCL professors and staffs.

I would like to give special thanks to Prof. Wei-Bin Zhang for the opportunity of the half-year research visit at the California PATH Program of the University of California, Berkeley, and all the support he has shown me throughout my research and career. I would also like to thank all PATH members and colleagues for fruitful and insightful discussions and contributions.

Ehsan Javanmardi

ジャワンマーディ　エッサン

List of publications

Journals:

1. **E. Javanmardi**, Y. Gu, M. Javanmardi, and S. Kamijo, “Factors to Evaluate Capability of Map for Vehicle Localization,” IEEE Access, (Submitted).

2. **E. Javanmardi**, Y. Gu, M. Javanmardi, and S. Kamijo, “Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area,” IATSS Research, May 2018.

3. M. Javanmardi, **E. Javanmardi**, Y. Gu, and S. Kamijo, “Towards High-Definition 3D Urban Mapping: Road Feature-Based Registration of Mobile Mapping Systems and Aerial Imagery,” Remote Sens., vol. 9, no. 10, p. 975, Sep. 2017.

International Conferences:

4. **E. Javanmardi**, M. Javanmardi, Y. Gu, and S. Kamijo, “Adaptive Resolution refinement of NDT Map Based on Localization Error Modeled by Map Factors,” to be presented in *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*, October 2018.

5. **E. Javanmardi**, M. Javanmardi, Y. Gu, and S. Kamijo, “Evaluation of Digital Map Ability for Vehicle Self-localization,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 437–442.

6. **E. Javanmardi**, M. Javanmardi, Y. Gu, and S. Kamijo, “Autonomous vehicle self-localization based on probabilistic planar surface map and multi-channel LiDAR in urban area,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.
   - ❖ (**Best student paper award**)

7. **E. Javanmardi**, M. Javanmardi, Y. Gu, and S. Kamijo, “Autonomous vehicle self-localization based on multilayer 2D vector map and multi-channel LiDAR,” in *2017 IEEE Intelligent Vehicles Symposium (IV), 2017*, pp. 437–442.

8. Z. Yue, **J. Ehsan**, J. Mahdi, G. Yanlei, and K. Shunsuke, “Towards ADAS Map: Automatic Extraction and Integration of Semantic Traffic Regulation from MMS Data,” *IEICE Technical Report; IEICE Tech. Rep.*, vol. 118, no. 79, pp. 7–12, Jun. 2018.

9. Mahdi Javanmardi, **Ehsan Javanmardi**, Yanlei Gu, and Shunsuke Kamijo, “A Novel Approach for Post-Calibration of Mobile Mapping Systems Using Intensity Reflection and Airborne Imagery,” Presented at *Transportation Research Board 96th Annual Meeting (TRB)*, Jan 2017.

10. Mahdi Javanmardi, **Ehsan Javanmardi**, Yanlei Gu, and Shunsuke Kamijo, "Automatic Calibration of 3D Mobile Laser Scanning using Aerial Surveillance Data for Precise Urban Mapping," in *IEEE Intelligent Vehicles Symposium (IV)*, June 2017.

11. Mahdi Javanmardi, **Ehsan Javanmardi**, Yanlei Gu, and Shunsuke Kamijo, "Precise Mobile Laser Scanning for Urban Mapping Utilizing 3D Aerial Surveillance Data," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, October 2017.

12. Mahdi Javanmardi, Yanlei Gu, **Ehsan Javanmardi**, Li-Ta Hsu, and Shunsuke Kamijo, "3D Building Map Reconstruction in Dense Urban Areas by Integrating Airborne Laser Point Cloud with 2D Boundary map," Presented at *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Nov 2015.

References

[1] P. Barrios, M. Adams, K. Leung, F. Inostroza, G. Naqvi, and M. E. Orchard, “Metrics for Evaluating Feature-Based Mapping Performance,” *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 198–213, Feb. 2017.

[2] M. Javanmardi, E. Javanmardi, Y. Gu, and S. Kamijo, “A Novel Approach for Post-Calibration of Mobile Mapping Systems Using Intensity Reflection and Airborne Imagery,” presented at the Transportation Research Board 96th Annual MeetingTransportation Research Board, 2017.

[3] M. Javanmardi, E. Javanmardi, Y. Gu, and S. Kamijo, “Towards High-Definition 3D Urban Mapping: Road Feature-Based Registration of Mobile Mapping Systems and Aerial Imagery,” *Remote Sens.*, vol. 9, no. 10, p. 975, Sep. 2017.

[4] M. Javanmardi, E. Javanmardi, Y. Gu, and S. Kamijo, “A Novel Approach for Post-Calibration of Mobile Mapping Systems Using Intensity Reflection and Airborne Imagery,” presented at the Transportation Research Board 96th Annual MeetingTransportation Research Board, 2017.

[5] 公昭乾, 昌英森川, 雅文橋本, 康平所谷, and 和彦高橋, “NDTスキャンマッチングに基づく車載レーザスキャナ観測点群の歪補正,” 日本機械学会論文集, vol. 83, no. 854, pp. 17-00061-17–00061, 2017.

[6] L. Cheng, Y. Wu, L. Tong, Y. Chen, and M. Li, “Hierarchical Registration Method for Airborne and Vehicle LiDAR Point Cloud,” *Remote Sens.*, vol. 7, no. 10, pp. 13921–13944, Oct. 2015.

[7] L. Yao *et al.*, “Registration of Vehicle-Borne Point Clouds and Panoramic Images Based on Sensor Constellations,” *Sensors*, vol. 17, no. 4, Apr. 2017.

[8] González-Jorge Higinio, Martínez Sánchez Joaquín, Díaz-Vilariño Lucía, Puente Iván, and Arias Pedro, “Automatic Registration of Mobile LiDAR Data Using High-Reflectivity Traffic Signs,” *J. Constr. Eng. Manag.*, vol. 142, no. 8, p. 04016022, Aug. 2016.

[9] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, “Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1356–1363.

[10] T. Kaminade, T. Takubo, Y. Mae, and T. Arai, “The generation of environmental map based on a NDT grid mapping -Proposal of convergence calculation corresponding to high resolution grid-,” in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, 2008, pp. 1874–1879.

[11] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan registration for autonomous mining vehicles using 3D-NDT,” *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.

[12] E. Einhorn and H. M. Gross, “Generic 2D/3D SLAM with NDT maps for lifelong application,” in *2013 European Conference on Mobile Robots (ECMR)*, 2013, pp. 240–247.

[13] L. Jun, L. Wei, D. Donglai, and S. Qiang, “Point cloud registration algorithm based on NDT

with variable size voxel," in *Control Conference (CCC), 2015 34th Chinese*, 2015, pp. 3707–3712.

[14] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "Normal distributions transform Monte-Carlo localization (NDT-MCL)," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 382–389.

[15] R. Cupec, E. K. Nyarko, D. Filko, A. Kitanov, and I. Petrović, "Global Localization Based on 3D Planar Surface Segments," *ArXiv13100314 Cs*, Oct. 2013.

[16] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, 2008, pp. 19–25.

[17] K. Yoneda, C. Yang, S. Mita, T. Okuya, and K. Muto, "Urban road localization by using multiple layer map matching and line segment matching," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 525–530.

[18] 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 *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3907–3912.

[19] U. Iqbal, J. Georgy, W. F. Abdelfatah, M. J. Korenberg, and A. Noureldin, "Pseudoranges error correction in partial gps outages for a nonlinear tightly coupled integrated system," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1510–1525, 2013.

[20] Q. Xu, X. Li, B. Li, X. Song, and Z. Cai, "A Reliable Hybrid Positioning Methodology for Land Vehicles Using Low-Cost Sensors," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 834–847, 2016.

[21] R. Yozevitch, B. Ben-Moshe, and A. Dvir, "GNSS Accuracy Improvement Using Rapid Shadow Transitions," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1113–1122, 2014.

[22] S. Zair, S. Le Hégarat-Mascle, and E. Seignez, "A-contrario modeling for robust localization using raw GNSS data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1354–1367, 2016.

[23] L. Heng, T. Walter, P. Enge, and G. X. Gao, "GNSS Multipath and Jamming Mitigation Using High-Mask-Angle Antennas and Multiple Constellations," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 741–750, 2015.

[24] D. Betaille, F. Peyret, M. Ortiz, S. Miquel, and L. Fontenay, "A new modeling based on urban trenches to improve GNSS positioning quality of service in cities," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 3, pp. 59–70, 2013.

[25] S. Miura, L.-T. Hsu, F. Chen, and S. Kamijo, "GPS error correction with pseudorange evaluation using three-dimensional maps," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3104–3115, 2015.

[26] V. Drevelle and P. Bonnifait, "Reliable positioning domain computation for urban navigation," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 3, pp. 21–29, 2013.

[27] Z. Wu, M. Yao, H. Ma, and W. Jia, “Improving accuracy of the vehicle attitude estimation for low-cost INS/GPS integration aided by the GPS-measured course angle,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 553–564, 2013.

[28] W. Hong, K. Choi, E. Lee, S. Im, and M. Heo, “Analysis of GNSS Performance Index Using Feature Points of Sky-View Image,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 889–895, 2014.

[29] N. M. Drawil, H. M. Amar, and O. A. Basir, “GPS localization accuracy classification: A context-based approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 262–273, 2013.

[30] K. Jo, M. Lee, and M. Sunwoo, “Road Slope Aided Vehicle Position Estimation System Based on Sensor Fusion of GPS and Automotive Onboard Sensors,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 250–263, 2016.

[31] E. I. Laftchiev, C. M. Lagoa, and S. N. Brennan, “Vehicle localization using in-vehicle pitch data and dynamical models,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 206–220, 2015.

[32] B. Ranft and C. Stiller, “The Role of Machine Vision for Intelligent Vehicles,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 8–19, Mar. 2016.

[33] H. Lategahn and C. Stiller, “Vision-only localization,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1246–1257, 2014.

[34] K. Jo, Y. Jo, J. K. Suhr, H. G. Jung, and M. Sunwoo, “Precise Localization of an Autonomous Car Based on Probabilistic Noise Models of Road Surface Marker Features Using Multiple Cameras,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3377–3392, 2015.

[35] S. Nedevschi, V. Popescu, R. Danescu, T. Marita, and F. Oniga, “Accurate ego-vehicle global localization at intersections through alignment of visual data with digital map,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 673–687, 2013.

[36] T. Mouats, N. Aouf, A. D. Sappa, C. Aguilera, and R. Toledo, “Multispectral stereo odometry,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1210–1224, 2015.

[37] C. Patruno, R. Marani, M. Nitti, T. D’Orazio, and E. Stella, “An Embedded Vision System for Real-Time Autonomous Localization Using Laser Profilometry,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3482–3495, 2015.

[38] S. Sivaraman and M. M. Trivedi, “Integrated lane and vehicle detection, localization, and tracking: A synergistic approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 906–917, 2013.

[39] A. A. Brown and S. N. Brennan, “Lateral Vehicle State and Environment Estimation Using Temporally Previewed Mapped Lane Features,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1601–1608, 2015.

[40] D. Cui, J. Xue, and N. Zheng, “Real-Time Global Localization of Robotic Cars in Lane Level via Lane Marking Detection and Shape Registration,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17,

no. 4, pp. 1039–1050, 2016.

[41] C. Rose, J. Britt, J. Allen, and D. Bevly, “An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2615–2629, 2014.

[42] N. Alam, A. Kealy, and A. G. Dempster, “Cooperative inertial navigation for GNSS-Challenged vehicular environments,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1370–1379, 2013.

[43] K. Lassoued, P. Bonnifait, and I. Fantoni, “Cooperative Localization with Reliable Confidence Domains between Vehicles sharing GNSS Pseudoranges Errors with no Base Station,” *IEEE Intell. Transp. Syst. Mag.*, 2016.

[44] M. Rohani, D. Gingras, and D. Gruyer, “A Novel Approach for Improved Vehicular Positioning Using Cooperative Map Matching and Dynamic Base Station DGPS Concept,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 230–239, 2016.

[45] K. Ansari, Y. Feng, and M. Tang, “A runtime integrity monitoring framework for real-time relative positioning systems based on GPS and DSRC,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 980–992, 2015.

[46] N. Alam, A. T. Balaei, and A. G. Dempster, “Relative positioning enhancement in VANETs: A tight integration approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 47–55, 2013.

[47] N. Alam and A. G. Dempster, “Cooperative positioning for vehicular networks: facts and future,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1708–1717, 2013.

[48] N. Alam, A. Kealy, and A. G. Dempster, “An INS-aided tight integration approach for relative positioning enhancement in VANETs,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1992–1996, 2013.

[49] H. Li, F. Nashashibi, and M. Yang, “Split covariance intersection filter: Theory and its application to vehicle localization,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1860–1871, 2013.

[50] H. Li and F. Nashashibi, “Cooperative multi-vehicle localization using split covariance intersection filter,” *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 2, pp. 33–44, 2013.

[51] G. D. Angelis, G. Baruffa, and S. Cacopardi, “GNSS/Cellular Hybrid Positioning System for Mobile Users in Urban Scenarios,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 313–321, Mar. 2013.

[52] M. Borenovic, A. Neskovic, and N. Neskovic, “Vehicle positioning using GSM and cascade-connected ANN structures,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 34–46, 2013.

[53] F. Moosmann and C. Stiller, “Velodyne SLAM,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 393–398.

[54] A. Y. Hata and D. F. Wolf, “Feature Detection for Vehicle Localization in Urban Environments Using a Multilayer LIDAR,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420–429, Feb.

2016.

[55] J. Choi, “Hybrid map-based SLAM using a Velodyne laser scanner,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 3082–3087.

[56] A. Y. Hata, F. S. Osorio, and D. F. Wolf, “Robust curb detection and vehicle localization in urban environments,” in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1257–1262.

[57] R. W. Wolcott and R. M. Eustice, “Visual localization within LIDAR maps for automated urban driving,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 176–183.

[58] Rongxing Li, “Mobile Mapping: An Emerging Technology for Spatial Data Acquisition,” *J. Am. Soc. Photogramm. REMOTE Sens.*, vol. 63, no. 9.

[59] M. J. Olsen, *Guidelines for the Use of Mobile LIDAR in Transportation Applications*. Transportation Research Board, 2013.

[60] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, “Beyond points: Evaluating recent 3D scan-matching algorithms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3631–3637.

[61] M. Magnusson, “The Three-Dimensional Normal-Distributions Transform --- an Efficient Representation for Registration, Surface Analysis, and Loop Detection,” *ResearchGate*, Dec. 2009.

[62] P. Biber and W. Strasser, “The normal distributions transform: a new approach to laser scan matching,” in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings*, 2003, vol. 3, pp. 2743–2748 vol.3.

[63] T. Kaminade, T. Takubo, Y. Mae, and T. Arai, “The generation of environmental map based on a NDT grid mapping -Proposal of convergence calculation corresponding to high resolution grid-,” in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1874–1879.

[64] M. Pauly and M. Gross, “Spectral Processing of Point-sampled Geometry,” in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2001, pp. 379–386.

[65] T. P. Kucner, M. Magnusson, and A. J. Lilienthal, “Where am I? An NDT-based prior for MCL,” in *2015 European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.

[66] M. Attia, Y. Slama, and M. A. Kamoun, “On Performance Evaluation of Registration Algorithms for 3D Point Clouds,” in *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*, 2016, pp. 45–50.

[67] M. Magnusson, N. Vaskevicius, T. Stoyanov, K. Pathak, and A. Birk, “Beyond points: Evaluating recent 3D scan-matching algorithms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3631–3637.

[68] M. Magnusson, R. Elsrud, L. Skagerlund, and T. Duckett, “3D Modelling for Underground Mining Vehicles,” in *In SimSafe 2005, Proceedings of the Conference on Modeling and Simulation for Public Safety*, 2005.

[69] T. Stoyanov, M. Magnusson, H. Almqvist, and A. J. Lilienthal, “On the accuracy of the 3D Normal Distributions Transform as a tool for spatial representation,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4080–4085.

[70] M. Magnusson, “The Three-Dimensional Normal-Distributions Transform --- an Efficient Representation for Registration, Surface Analysis, and Loop Detection,” 2009.

[71] “CSF: Ground filtering of point cloud based on Cloth Simulation by Wuming Zhang - Research Project on ResearchGate,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/project/CSF-Ground-filtering-of-point-cloud-based-on-Cloth-Simulation. [Accessed: 03-May-2017].

[72] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for Point-Cloud Shape Detection,” *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.

[73] O. Vysotska and C. Stachniss, “Improving SLAM by Exploiting Building Information from Publicly Available Maps and Localization Priors,” *PFG – J. Photogramm. Remote Sens. Geoinformation Sci.*, vol. 85, no. 1, pp. 53–65, Feb. 2017.

[74] H. Merzić, E. Stumm, M. Dymczyk, R. Siegwart, and I. Gilitschenski, “Map quality evaluation for visual localization,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3200–3206.

[75] S. Thompson and S. Kagami, “Evaluating 3D Polygon Maps for Mobile Robot Localisation,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 2396–2401.

[76] H. Almqvist, M. Magnusson, T. P. Kucner, and A. J. Lilienthal, “Learning to detect misaligned point clouds,” *J. Field Robot.*, vol. 0, no. 0.

[77] A. Makadia, A. Patterson, and K. Daniilidis, “Fully Automatic Registration of 3D Point Clouds,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, 2006, vol. 1, pp. 1297–1304.

[78] L. Silva, O. R. P. Bellon, and K. L. Boyer, “Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 762–776, May 2005.

[79] D. Campbell and M. Whitty, “Metric-based detection of robot kidnapping,” in *2013 European Conference on Mobile Robots*, 2013, pp. 192–197.

[80] “Efficient DOP Calculation for GPS with and without Altimeter Aiding | The Journal of Navigation |Cambridge Core.” [Online]. Available: https://www.cambridge.org/core/journals/journal-of-navigation/article/efficient-dop-calculation-for-gps-with-and-without-altimeter-

aiding/8FD31C94F27E922CD1401FD3EEA5BE70. [Accessed: 22-Feb-2018].

[81] A. Gressin, C. Mallet, and N. David, “Improving 3d LIDAR Point Cloud Registration Using Optimal Neighborhood Knowledge,” *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 3, pp. 111–116, Jul. 2012.

[82] M. A. Savelonas, I. Pratikakis, and K. Sfikas, “Fisher encoding of differential fast point feature histograms for partial 3D object retrieval,” *Pattern Recognit.*, vol. 55, no. Supplement C, pp. 114–124, Jul. 2016.

[83] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D Registration,” in *in In Proceedings of the International Conference on Robotics and Automation (ICRA*, 2009.

[84] 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 *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3907–3912.