Road features detection and sparse map-based vehicle localization in urban environments

Alberto Yukinobu Hata

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Alberto Yukinobu Hata

Road features detection and sparse map-based vehicle localization in urban environments

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. FINAL VERSION

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
February 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/ USP,
com os dados fornecidos pelo(a) autor(a)

H634r Hata, Alberto Yukinobu
Road features detection and sparse map-based
vehicle localization in urban environments / Alberto
Yukinobu Hata; orientador Denis Fernando Wolf. - São
Carlos - SP, 2017.
100 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional)
- Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. Vehicle localization. 2. Gaussian process
occupancy map. 3. occupancy grid map. 4. curb
detection. 5. road marking detection. 6. Monte Carlo
localization. I. Wolf, Denis Fernando, orient. II.
Título.

Alberto Yukinobu Hata

Detecção de características de rua e localização de veículos em ambientes urbanos baseada em mapas esparsos

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. VERSÃO REVISADA

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Denis Fernando Wolf

USP – São Carlos
Fevereiro de 2017

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Prof. Denis Fernando Wolf for the support provided during the Ph.D. His patience and thorough knowledge in the robotics field were essential for the development of this work. His feedback during the work contributed a lot to my personal growth.

The members of Mobile Robotics Laboratory (LRM-ICMC) also helped immensely to solve varied difficulties along the Ph.D., as well as to execute the numerous experiments through the autonomous vehicle prototype. Moreover, without the materials available at LRM, it would not be possible to conduct this project. I also acknowledge the employees and the professors of Institute of Mathematical and Computer Science for their support along these years.

I'm also grateful to Prof. Fábio Ramos of the University of Sydney who supported me during my internship and helped me to understand Bayesian learning field and Gaussian process theory. His students were always available to clarify particular subjects related to robotics field and Gaussian Process mapping. The assistance obtained by members of Australian Centre for Field Robotics at the University of Sydney were also fundamental during the internship.

The financial support provided by FAPESP (process no. 2012/02354-1 and process no. 2014/09096-3) and CAPES (process no. DS-6344422/D) were essential to buy equipment and participate in important conferences of autonomous vehicles and robotics field.

Last but not least, I thank my family to the incentive they gave me along the Ph.D. and to understand my absence at important moments. I especially thank my girlfriend Marcilyanne Gois that followed me during all the Ph.D. Without her patience, incentive and comprehension, it would not be possible to accomplish this work.

RESUMO

HATA, A. Y.. Road features detection and sparse map-based vehicle localization in urban environments. 2017. 100 f. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

No contexto de veículos autônomos, a localização é um dos componentes fundamentais, pois possibilita tarefas como ultrapassagem, direção assistida e navegação autônoma. A presença de edifícios e o mau tempo interferem na recepção do sinal de GPS que consequentemente dificulta o uso de tal tecnologia para a localização de veículos dentro das cidades.

Alternativamente, a localização com suporte aos mapas vem sendo empregada para estimar a posição sem a dependência do GPS. Nesta solução, a posição do veículo é dada pela região em que ocorre a melhor correspondência entre o mapa do ambiente e a leitura do sensor.

Antes da criação dos mapas, características dos ambientes devem ser extraídas a partir das leituras dos sensores. Dessa forma, guias e sinalizações horizontais têm sido largamente utilizados para o mapeamento. Entretanto, métodos de mapeamento urbano geralmente necessitam de repetidas leituras do mesmo lugar para compensar as oclusões. A construção de representações precisas dos ambientes é essencial para uma adequada associação dos dados dos sensores com o mapa durante a localização.

De forma a evitar a necessidade de um processo manual para remover obstáculos que causam oclusão e áreas não observadas, propõe-se um método de localização de veículos com suporte aos mapas construídos a partir de observações parciais do ambiente. No sistema de localização proposto, os mapas são construídos a partir de guias e sinalizações horizontais extraídas a partir de leituras de um sensor multicamadas. As guias podem ser detectadas mesmo na presença de veículos que obstruem a percepção das ruas, por meio do uso de regressão robusta. Na detecção de sinalizações horizontais é empregado o método de limiarização por Otsu que analisa dados de reflexão infravermelho, o que torna o método insensível à variação de luminosidade.

Dois tipos de mapas são empregados para a representação das guias e das sinalizações horizontais: mapa de grade de ocupação (OGM) e mapa de ocupação por processo Gaussiano (GPOM). O OGM é uma estrutura que representa o ambiente por meio de uma grade reticulada. O GPOM é uma representação contínua que possibilita a estimação de áreas não observadas. O método de localização por Monte Carlo (MCL) foi adaptado para suportar os mapas construídos. Dessa forma, a localização de veículos foi testada em MCL com suporte ao OGM e MCL com suporte ao GPOM. No caso do MCL baseado em GPOM, um novo modelo de verossimilhança baseado em função densidade probabilidade de distribuição multi-normal é proposto.

Experimentos foram realizados em ambientes urbanos reais. Mapas do ambiente foram gerados a partir de dados de laser esparsos de forma a verificar a reconstrução de áreas não observadas.

O sistema de localização foi avaliado por meio da comparação das posições estimadas com um GPS de alta precisão. Comparou-se também o MCL baseado em OGM com o MCL baseado em GPOM, de forma a verificar qual abordagem apresenta melhores resultados.

Palavras-chave: Localização de veículos, Mapa de ocupação de processo Gaussiano, Mapa de grade de ocupação, detecção de guia, detecção de sinalização horizontal, localização de Monte Carlo.

ABSTRACT

HATA, A. Y.. Road features detection and sparse map-based vehicle localization in urban environments. 2017. 100 f. Doctoral dissertation (Doctorate Candidate Program in Computer Science and Computational Mathematics) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Localization is one of the fundamental components of autonomous vehicles by enabling tasks as overtaking, lane keeping and self-navigation. Urban canyons and bad weather interfere with the reception of GPS satellite signal which prohibits the exclusive use of such technology for vehicle localization in urban places. Alternatively, map-aided localization methods have been employed to enable position estimation without the dependence on GPS devices. In this solution, the vehicle position is given as the place that best matches the sensor measurement to the environment map.

Before building the maps, features of the environment must be extracted from sensor measurements. In vehicle localization, curbs and road markings have been extensively employed as mapping features. However, most of the urban mapping methods rely on a street free of obstacles or require repetitive measurements of the same place to avoid occlusions. The construction of an accurate representation of the environment is necessary for a proper match of sensor measurements to the map during localization.

To prevent the necessity of a manual process to remove occluding obstacles and unobserved areas, a vehicle localization method that supports maps built from partial observations of the environment is proposed. In this localization system, maps are formed by curb and road markings extracted from multilayer laser sensor measurements. Curb structures are detected even in the presence of vehicles that occlude the roadsides, thanks to the use of robust regression. Road markings detector employs Otsu thresholding to analyze infrared remittance data which makes the method insensitive to illumination.

Detected road features are stored in two map representations: occupancy grid map (OGM) and Gaussian process occupancy map (GPOM). The first approach is a popular map structure that represents the environment through fine-grained grids. The second approach is a continuous representation that can estimate the occupancy of unseen areas. The Monte Carlo localization (MCL) method was adapted to support the obtained maps of the urban environment. In this sense, vehicle localization was tested in an MCL that supports OGM and an MCL that supports GPOM. Precisely, for MCL based on GPOM, a new measurement likelihood based on multivariate normal probability density function is formulated.

Experiments were performed in real urban environments. Maps were built using sparse laser data to verify the reconstruction of non-observed areas. The localization system was evaluated

by comparing the results with a high precision GPS device. Results were also compared with localization based on OGM.

Key-words: Vehicle localization, Gaussian process occupancy map, occupancy grid map, curb detection, road marking detection, Monte Carlo localization.

LIST OF FIGURES

Figure 1	– Representation of concentric rings returned by the multilayer laser sensor when the beams hit a flat surface.	34
Figure 2	– Rings returned from laser measurement compress with rates proportional to surface slope. In this laser frame, different obstacles can be visually distinguished, such as a wall (yellow area), grass (green area), street (blue area) and curbs (red area).	34
Figure 3	– Circular grid representation used to manage laser point cloud during curb detection. Each circular row stores point cloud belonging to a single ring. Cells store the average values of those points that fall in.	35
Figure 4	– Curb detection in an occluded road. (a) Occlusion caused by a car on the street. False-positives (yellow points) are removed by estimating the road shape (blue line) from other detected points. (b) Performance comparison of curve fitting methods. Axes are in meter unit.	37
Figure 5	– Curb detection steps in an occluded scene in a steep street. This scenario illustrates the effect of gradient and regression filter to remove false-positives generated by ring compression.	38
Figure 6	– Road marking detection based on infrared intensity analysis. (a) Raw intensity data returned by the laser sensor artificially colored according to the intensity value. Note that there are a substantial difference between road marking and asphalt points. (b) Intensity histogram of points belonging to a single ring represented in the left image. Two classes are present, grouped by the peaks of the function in red.	39
Figure 7	– Functions randomly sampled from the GP distribution and the corresponding uncertainty (gray area). (a) Before the inclusion of the training dataset, the uncertainty is large and consequently the space of function is wide. (b) After adding training samples (plus symbol), the uncertainty reduces, in special around the training points. The space of functions starts to get a delineated shape. Figures extracted from (RASMUSSEN; WILLIAMS, 2005).	45
Figure 8	– Effects of different parameters ℓ on the GP regression. Curve smoothness and the confidence changes significantly. Blue points, plus symbol and gray area denotes query values, training data and variance, respectively. Figures extracted from (RASMUSSEN; WILLIAMS, 2005).	46

Figure 9 – Representation of a single laser scan using discrete (OGM) and continuous (GPOM) representations. The training (observation) data set was overlapped in the GPOM images (yellow corresponds to free space and green to occupied space). Blue and red colors are associated with free and occupied regions, respectively. Axes scale is in meters.	47
Figure 10 – GP regression using point and line observations. Estimated mean and variance are represented by red dot and the colored area.	49
Figure 11 – Example of information theory data compression applied in a laser dataset. Red segments represent the unnecessary laser beams removed from the original dataset. In this example, 1789 of 2941 beams were discarded (60.38% reduction), though visually the dataset reduction is practically unnoticeable.	51
Figure 12 – Grouping subsets of training dataset using k-means. The clusters are bounded by rectangles and the centroid represented by the black star.	52
Figure 13 – Heuristics to enable GP mapping of large areas. (a) workflow to associate the training data to the clusters and (b) gating network to set the GP expert to infer the occupancy of each test point.	53
Figure 14 – Example of MCL executing inside an office environment containing several rooms. (a) At the beginning of MCL, particles are uniformly distributed over the environment. (b) In ambiguous situation (measurements have been similar at more than one location), the particle set keep multiple hypothesis of the position. (c) The particle set concentrates at a single location after discarding the incorrect hypotheses by gathering more sensor measurements in the subsequent MCL iterations. Source: (DELLAERT et al., 1999).	57
Figure 15 – Example of the range measurement likelihood field representation of an arbitrary environment. Darker regions corresponds to smaller likelihood values.	58
Figure 16 – Environment mapping diagram. The mapping process takes the GNSS coordinates and laser sensor as inputs and generates GPOM from curb data and OGM from road marking data. Two mapping approaches were employed to represent curb features.	63
Figure 17 – Vehicle localization diagram. The MCL method receives laser data (curb and road marking point clouds), map and wheel encoder data to generate the position distribution posterior. The curb map leads to two localization approaches.	64
Figure 18 – Maps built from sparse laser data of Seattle and Bicocca datasets. (a–b) Ground truth, (c–d) OGM and (e–f) GPOM. Blue color is associated to free areas and red color to occupied areas. Axes are in meters.	69

Figure 19 – Maps built from sparse laser data of Bicocca dataset. (a) Reference map, (b) laser data reduction trough information theoretic method (red lines denotes the removed beams) and data clustering trough k-means ($k = 15$), (c) OGM and (d) GPOM modeled by 15 GP experts. Blue color is associated to free areas and red color to occupied areas. Axes are in meters.	69
Figure 20 – Global localization results of the indoor datasets along the test trajectory. The estimated position is denoted by blue lines and the ground truth pose is denoted by red lines. Euclidean error and particle variance is also plotted along the robot path which are represented by blue circles and green ellipses, respectively. Axes are in meters.	70
Figure 21 – Carina II autonomous vehicle prototype equipped with Velodyne HDL-32E sensor on the top (in gray color).	72
Figure 22 – Curb detection before (a–c) and after (d–e) parameter tuning of three scenarios: left turn (a, d), straight road (b, e) and right turn (c, f). Red points denote the detected curb points and the blue lines denote the LTS regression curve. The number of detected curb points increases after the parameter tuning.	73
Figure 23 – Curb detection in different situations. (a) Cars on both street sides. (b) Two cars parked on the road side. (c) Two distant cars on the road side. Red points denote the detected curb points and the blue lines denote the LTS regression curve.	74
Figure 24 – (a)–(b) Factory original intensity. (c)–(d) Calibrated intensity which leads to a better contrast between asphalt and road markings. The point cloud was artificially colored according to the remittance intensity value for a better visualization.	75
Figure 25 – Road marking detected by a modified version of Otsu method. (a)–(d) Single frame result. (e)–(g) Accumulated results from consecutive frames. Blue lines represent the road boundaries estimated by the curb detector and yellow points correspond to the detected road marking.	76
Figure 26 – Intensity histogram of the LIDAR points that intercept road marking. The dashed line represents the obtained threshold.	76
Figure 27 – Environment maps used in the localization experiments. Track 1 is 770 m long and Track 2 is 1749 m long.	77
Figure 28 – Training data set used to build GPOM and OGM. (a)–(b) Clustered mapping (training) dataset distinguished by a color. (c)–(d) OGM configured with 0.10 cm resolution. Blue and red colors are related to free and occupied areas, respectively. Axes are in meters.	78

Figure 29 – Output values of the GP mapping. The GPOM corresponds to the squashed mean values. (a)–(b) Mean value. (c)–(d) Variance Value. (e)–(f) Squashed mean obtained from mean and variance values. Blue and red colors are associated to free and occupied areas, respectively. Axes are in meters. . . .	79
Figure 30 – Amplified regions of Track 1 maps. Blue and red colors are associated to free and occupied areas, respectively. Axes are in meters.	80
Figure 31 – Amplified regions of Track 2 maps. Blue and red colors are associated to free and occupied areas, respectively. Axes are in meters.	81
Figure 32 – Occupancy grid maps with 0.10 m resolution containing road marking data. Axes are in meters.	82
Figure 33 – Zoomed regions of road marking map of Track 1.	83
Figure 34 – Zoomed regions of road marking map of Track 2. Axes are in meters.	83
Figure 35 – Comparison of position sensors output obtained from Track 1 path. Taking the RTK GNSS (red points) as a reference, wheel encoder (blue points) accumulates position error as the traveled distance increases and the low-cost GPS (green points) gets shifted in parts of the track. Axes are in meters. . .	84
Figure 36 – Laser likelihood field with 0.10 m resolution build using OGM containing curb data and OGM containing road marking data. Axes are in meters. . . .	85
Figure 37 – Localization errors superimposed in the traveled path of Track 1. (a)–(b) MCL-OGM using curb and both curb and road marking observations. (c)–(d) MCL-GPOM using curb and both curb and road marking observations. Axes are in meters.	86
Figure 38 – Localization errors of Track 1. First row: localization using curb observations. Second row: localization using both curb and road marking observations. Values closer to 0 are associated to lower errors.	87
Figure 39 – Localization errors superimposed in the traveled path of Track 2. (a)–(b) MCL-OGM using curb and both curb and road marking observations. (c)–(d) MCL-GPOM using curb and both curb and road marking observations. Axes are in meters.	89
Figure 40 – Localization errors of Track 2. First row: localization using curb observations. Second row: localization using both curb and road marking observations. Values closer to 0 are associated to lower errors.	90

LIST OF TABLES

Table 1 – Comparison of the performance of different fitting methods for the estimation of the curb model.	37
Table 2 – Details of the datasets used in the indoor mapping and localization experiments.	68
Table 3 – Numerical results for global localization in indoor environments. Absolute average longitudinal (x), lateral (y), Euclidean (xy) and angular (θ) errors of Seattle, Belgioioso and Bicocca scenarios.	70
Table 4 – Details of the tracks used in the mapping and localization experiments.	77
Table 5 – Position errors of the low-cost GNSS and rear wheel encoders after traveled a single loop of Track 1. Axes are in meters.	84
Table 6 – Track 1 localization errors. Absolute average longitudinal (x), lateral (y), Euclidean (xy) and angular (θ) errors of Tracks 1.	87
Table 7 – Track 2 localization errors. Absolute average longitudinal (x), lateral (y), euclidean (xy) and angular (θ) errors of Tracks 1.	90

LIST OF ABBREVIATIONS AND ACRONYMS

DGPS		Differential GPS
GIS		Geographic Information System
GNSS		Global Navigation Satellite System
GP		Gaussian Process
GPOM	...	Gaussian Process Occupancy Map
GPS		Global Positioning System
ICP		Iterative Closest Point
MCL		Monte Carlo Localization
MCL-GPOM		MCL method based on GPOM
MCL-OGM		MCL method based on OGM
MSE		Mean Squared Error
OGM		Occupancy grid map
RANSAC	.	Random Sample Consensus
RTK		Real Time Kinematic
SLAM	...	Simultaneous Localization and Mapping
UTM		Universal Transverse Mercator

CONTENTS

1	INTRODUCTION	21
1.1	Objective	24
1.2	Contributions	24
1.3	Limitations	25
1.4	Structure of the Thesis	25
2	RELATED WORKS	27
2.1	Road Feature Detection	28
2.1.1	Curb Detection	28
2.1.2	Road Marking Detection	29
2.2	Mapping	30
2.3	Localization	31
2.4	Final Remarks	32
3	ROAD FEATURE DETECTION	33
3.1	Curb Detection	33
3.1.1	Gradient Filter	36
3.1.2	Occlusion Removal	36
3.1.3	Parameter Tuning	38
3.2	Road Marking Detection	38
3.3	Final Remarks	41
4	ENVIRONMENT MAPPING	43
4.1	Occupancy Grid Mapping	43
4.2	Gaussian Process	44
4.3	Gaussian Process Occupancy Mapping	46
4.3.1	Point and Line Observations	46
4.3.2	Covariance for Line Observations	47
4.3.2.1	Integral Kernel	47
4.3.3	Occupancy Prediction for New Point Observations	49
4.4	GP Regression Example	49
4.5	Gaussian Process Occupancy Mapping	50
4.5.1	Updating Inverse Matrix Through Cholesky Factor	50
4.5.2	Information Theoretic Compression of Training Data	51

4.5.3	Mixture of Gaussian Processes	52
4.5.4	Combining Heuristics for GPOM	53
4.5.5	Mathematical Optimization	53
4.6	Final Remarks	54
5	MONTE CARLO LOCALIZATION	55
5.1	General Algorithm	55
5.2	Likelihood for Occupancy Grid Map	57
5.3	Likelihood for Gaussian Process Occupancy Map	58
5.4	Multimap Likelihood	60
5.5	Position Prior	61
5.6	MCL Variations	61
5.7	Localization in Urban Environments	61
5.7.1	Urban Environment Mapping	62
5.7.2	Vehicle Localization	63
5.8	Final Remarks	65
6	EXPERIMENTS AND RESULTS	67
6.1	Experiments in Indoor Environments	67
6.2	Experiments in Urban Environments – Setup	71
6.3	Road Features Detection	73
6.3.1	Curb Detection	73
6.3.2	Road Marking Detection	75
6.4	Urban Environment Mapping	76
6.4.1	Curb mapping	77
6.4.2	Road marking mapping	82
6.5	Vehicle Localization in Urban Environments	83
6.5.1	MCL using OGM	85
6.5.2	MCL using GPOM	88
6.5.3	Comparison of MCL-GPOM and MCL-OGM Approaches	91
6.6	Final Remarks	91
7	CONCLUSION AND FUTURE WORKS	93
7.1	Future Works	94
7.2	Publications Derived from this Work and Contributions to Other Works	94
	REFERENCES	97

INTRODUCTION

The considerable expansion of car fleet along the years affected directly in the increase of the traffic accidents. In a global scenario, traffic accidents are the 9th most common death cause¹ and Brazil is the country with the 6th highest mortality rate caused by traffic accidents². Besides that, statistics show that the number of vehicles in Brazil increases every year and has reached in 2011 the 4th biggest vehicle fleet in the world³. For this reason, car companies, and robotics field researchers have been invested in partial or total automation of the vehicles, with the purpose of helping the motorists to drive safely on the streets.

Autonomous cars (a.k.a. self-driving cars) are comprised of fundamental systems, such as obstacle detection, navigation, and localization. Localization plays an important role, as it determines the vehicle position in the environment. Based on localization information, we can compute the position of an obstacle and perform the path planning. Therefore, the quality of the localization system affect the overall functionality of autonomous vehicles.

A straightforward way to get the vehicle position is by Global Navigation Satellite System (GNSS) devices, also popularly known as Global Positioning System (GPS) devices. These devices can calculate the geographic coordinates (latitude, longitude, and altitude) through the triangulation of a set of satellites. Although these devices are relatively cheap, the measurement error can reach 15 m⁴. A variation of GNSS, named Differential GPS (DGPS), uses a ground fixed antenna to get higher precision during triangulation, thus reducing the position errors to between 30 to 60 cm. This error is decreased to millimeter scale when the Real Time Kinematic (RTK) solution is used (WÜBBENA et al., 2001). However, DGPS and RTK technologies are most of the times expensive and consequently, infeasible for commercial application.

Despite the practical solution to vehicle localization is the use of high precision GNSS

¹Source: WHO, May 2014 (The 10 leading causes of death in the world, 2000 and 2012).

²Source: Instituto Sangari, April 2012 (Mapa da Violência 2012. Acidentes de Trânsito).

³Source: Fenabrave, Abril de 2012 (Desempenho da Distribuição Automotiva no Brasil).

⁴Position accuracy of Garmin 18x GPS. Source: http://static.garmincdn.com/pumac/GPS_18x_Tech_Specs.pdf

devices, there are external factors that make the exclusive dependency on these sensors dangerous, especially inside cities. The presence of trees, buildings, tunnels, bridges and also the cloudy weather may difficult the reception of the satellite signal, which can decrease the position accuracy or even making the sensor unusable. In critical applications such as autonomous vehicles, the absence of a localization system with high reliability and availability can conduct it to undesired places and induce to traffic accidents. Moreover, DGPS and RTK require the constant communication to a stationary antenna which may also difficult the installation of this antenna in large areas (e.g. cities).

To overcome the disadvantages of GNSS devices, featured autonomous vehicles of the literature ([THRUN et al., 2006](#); [URMSON et al., 2008](#)) make use of environment maps created beforehand to aid in the localization. Through this process, the position is estimated even in the absence of the satellite signal ([LUETTEL; HIMMELSBACH; WUENSCH, 2012](#)). In map-aided localization, the exteroceptive sensor⁵ readings are matched with the map to estimate the vehicle position. The position is given by the place that best match the sensor measurements to the map.

Metric and topological map structures are commonly used for localization. Metric maps store geometric information of the environment and are represented by a grid whose cells indicate the probability of the presence of an obstacle (i.e. probability of occupancy). Laser sensors are widely employed to build metric maps, because of the geometric information that they return. In the other hand, topological maps store contextual information and are usually represented by a graph whose edges and vertices denote links between places. Commonly, metric maps are preferred to topological maps because of the superior precision and relative facility to implement ([THRUN; BURGARD; FOX, 2005](#)). Occupancy grid map (OGM) is a popular type of metric map employed in autonomous vehicles field ([MONTEMERLO et al., 2008](#)).

Structures that are commonly present in urban scenarios and that barely suffer from drastic changes along the time are preferred to compose the environment maps. For instance, building facades, curbs and road markings have been widely employed as map features ([HILLEL et al., 2012](#)). Despite the existence of diverse methods for urban feature extraction, many of them assume the absence of occluding obstacles in the scenario ([LEVINSON et al., 2011](#); [QIN et al., 2012](#)), though this situation difficulty occurs.

As a consequence, repetitive measurements of the same place at different times and using high-resolution sensors are necessary to ensure complete observation of the environment. Another commonly employed solution is the use of third-party maps that provide high-definition and accurate representations. These workarounds are indispensable, because the incorrect detection of urban features may result in an inaccurate representation of the environment and therefore, interfere the matching of sensor measurements to the map during localization.

⁵Devices that extracts data from the environment, for example, laser sensors, cameras, and radars.

Alternatively, this work proposes the development of a vehicle localization method that can deal with maps generated from partial or incomplete observations of the environment. Therefore, redundant sensor measurements and dense sensors are dispensed during the mapping process. The proposed localization system relies on two urban features to build the environment map: curbs and road markings. These structures have been successfully employed in localization problems (URMSON et al., 2008; MONTEMERLO et al., 2008; LUETTEL; HIMMELSBACH; WUENSCH, 2012). The proposed curb detector relies on robust regression which makes the method insensitive to the presence of obstacles on the street (e.g. cars and pedestrians). Regarding road marking detector, it can extract any type of horizontal signalization (e.g. pedestrian crossing, lane marking and dashed marking) and is insensitive to rotation.

The environment perception relies only on a multilayer laser sensor. This sensor was adopted due to the possibility to retrieve metric data and infrared remittance intensity data (used to detect reflective objects, such as road markings). Moreover, the sensor measurements are less sensitive to external influence (e.g. shadow, lack, and excess of light) compared to cameras.

Two map structures are employed to reconstruct the environment from the detected road features: OGM and Gaussian Process Occupancy Map (GPOM). The first one is a grid-based representation that can capture fine details of the environment and the second one is a continuous representation that estimates the occupancy of unseen areas through Gaussian Process (GP) inference.

The localization method is based on Monte Carlo Localization (MCL) algorithm, because of its capability to deal with ambiguous situations and noisy sensor readings (FOX et al., 2001). The MCL method is derived from Particle Filter theory which uses a set of particles to approximate non-linear functions. Each particle is associated with a "weight" value that corresponds to the likelihood to represent the actual position. The MCL method was adapted to work with OGM and GPOM built from urban features. As the integration between OGM and MCL is known in the literature, a new likelihood function is designed for MCL to support GPOM representation.

In summary, the proposed localization system initially extracts curbs and road markings from point cloud data. The curb detector can operate even in the presence of other obstacles occluding the roadsides and road marking detector can detect any type of horizontal signalization, independently of the illumination condition. Rather than employing curb features to represent exclusively road boundaries, the navigable area was also reconstructed by generating a simulated laser scan data (single beam laser) from detected curb points. Additionally, laser data was intentionally downsampled to produce sparse measurements which mimic the presence of unseen areas. OGM and GPOM are evaluated to reconstruct the urban environment. GPOM has an inherent property of estimating the occupancy of unseen areas, while OGM presents some deficiency to deal with sparse measurements. However, GPOM has much higher computational cost compared to OGM. Road markings were solely represented through OGM, as it was used

as auxiliary information for vehicle localization. To check the effect of mapping approach on vehicle localization, MCL method based on OGM (MCL-OGM) and MCL based on GPOM (MCL-GPOM) were tested. For MCL-GPOM a new measurement likelihood function based on multivariate normal probability density function was employed.

Except for preliminary experiments in indoor scenarios, all experiments were performed in urban streets through Carina II autonomous vehicle prototype. For evaluation, the proposed localization method was compared with position obtained by a millimeter precision GNSS device enabled with RTK. Further, a comparison between OGM and GPOM for localization task was also conducted.

1.1 Objective

Localize a vehicle through map-aided localization using maps constructed from partial, incomplete or sparse laser sensor measurements gathered from urban environments.

The specific objectives of this work are:

- Detect curbs from point cloud data, considering the presence of obstacles occluding the roadsides.
- Detect road markings from point cloud data that contains infrared intensity data.
- Reconstruct the urban environment through occupancy grid mapping using the detected curb and road marking data.
- Reconstruct the urban environment through Gaussian process occupancy mapping using detected curb data.
- Localize the vehicle through MCL using the generated OGM.
- Devise a new likelihood model to enable localization through MCL using GPOM and localize the vehicle through this solution.

1.2 Contributions

The map-aided localization using map built from partial or incomplete observations of the environment enables a localization solution that: (i) does not rely on dense or high-resolution sensors, thus, favorable to low cost sensors; (ii) does not require redundant sensor measurements of the same location to avoid unobserved areas and occluded areas, consequently demanding fewer time for environment data gathering; (iii) uses GP mapping to estimate the occupancy of previously unseen areas; and (iv) is more accurate than localization approaches that employs discrete maps such as OGM (verified after experiments).

Regards to the road feature detection methods, the curb detector proposes a novel approach that: (i) uses ring compression information (returned by Velodyne laser sensors) and (ii) is robust to the presence of obstacles that occludes the sensor field of view to roadsides. The road marking detector: (i) makes use of infrared intensity information which makes less susceptible to illumination influence, compared to cameras; (ii) can detect any type of horizontal signalization, independent of the shape, color or position; and (iii) employs an adapted version of Otsu thresholding for a simple but efficient segmentation of the road markings.

1.3 Limitations

The proposed localization system does not deal with dynamic obstacles, therefore the generated maps are static. The localization is tested only on streets that have curbs and properly paved. Finally, the computing cost of the localization system is not taken into account.

1.4 Structure of the Thesis

The monograph is comprised of the following chapters:

- [Chapter 2](#): Presents a literature review of curb detection, road marking detection, environment mapping and localization methods.
- [Chapter 3](#): Describes curb detection and road marking detection developed in this work.
- [Chapter 4](#): Describes occupancy grid mapping, Gaussian process and Gaussian process occupancy mapping. Adaptations made in the mapping method to enable representation of large areas are also described.
- [Chapter 5](#): Describes Monte Carlo localization method and adaptations made to support Gaussian process occupancy map. Details of the proposed vehicle localization system that matches curb and road marking observations with the environment maps are also presented.
- [Chapter 6](#): Reports performed experiments for curb detection, road marking detection, urban environment mapping and vehicle localization. Evaluations and obtained results are also presented.
- [Chapter 7](#): Concludes the proposed vehicle localization method, presents future works and publications derived from this thesis.

RELATED WORKS

The localization system of early autonomous vehicles, such as project Prometheus and ALVINN ([TURK et al., 1988](#)), relied on information provided by sensors such as wheel encoders, Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU). Differently, vehicles that took part of DARPA Urban Challenge in 2007 used in addition to these sensors, maps of the environment to increase the localization accuracy ([THRUN et al., 2006](#)). These maps contained metric and semantic information of the road and the traffic signs. Therefore, the localization is performed by matching the sensor information to the environment map. This solution is referred as map-aided localization and can estimate the vehicle position with centimeter accuracy ([LEVINSON et al., 2011](#); [THRUN et al., 2006](#)).

Similarly, the current project uses maps for vehicle localization. However, we adopted continuous occupancy maps which are robust to unobserved areas and do not require environment discretization. The proposed localization system is formed by three fundamental components: road feature detection, environment mapping, and localization. These components are present in the most of the existing autonomous vehicles ([THRUN et al., 2006](#); [LEVINSON et al., 2011](#); [QIN et al., 2012](#)).

Road feature detection is responsible for extracting relevant information from roads. Particularly, two features were taken into account in this work: road curbs and road markings. Environment mapping is responsible for constructing the occupancy maps from the detected road features. Finally, localization associates the sensor measurement with the environment map to estimate the vehicle position.

The following sections present the bibliographic review of each component of the proposed localization system. As the proposed localization system relies on laser sensor for perception, only those works that adopt this sensor were covered.

2.1 Road Feature Detection

Curb and horizontal signalization are important structures in urban environments that enable safe guidance in the streets. There are articles describing the application of these elements for driving assistance, such as lane keeping, lane departure warning, and overtaking (HILLEL et al., 2012). Besides that, others works demonstrated the feasibility of using these structures for localization (THRUN et al., 2006; KAMMEL; PITZER, 2008; MONTEMERLO et al., 2008; QIN et al., 2012).

According to (LEVINSON et al., 2011), the first step of urban feature detection consists in removing the laser sensor readings that are not near the street level. This operation is straightforward, given the geometric property of the laser data. Some authors extract the road surface by simply discarding laser points higher than a certain value and others apply Random Sample Consensus (RANSAC) method to extract laser points that fit a plane model (HILLEL et al., 2012). After this process, road feature detection methods are performed in the remaining laser data.

The following sections describe the main works about curb and road marking detection based on laser sensor.

2.1.1 Curb Detection

Curbs are traffic structures with few centimeters height, commonly present at both sides of the street to delimit the navigable area. Considering the geometrical attribute of curbs, the use of laser sensor is a natural choice, as this sensor returns tridimensional data.

The majority of the curb detection methods is based on the analysis of the height variation of adjacent laser points situated near the street level (HILLEL et al., 2012). Following this idea, (ZHANG, 2010) applies the differential convolution in the height values of laser data to detect steep height changes. Those laser points associated with convolution result higher than a predefined value are classified as curbs. (QIN et al., 2012) included laser beam angle information to restrict the differential calculation to laser beams that potentially hit a curb. The main drawback of these approaches is the sensibility to the presence of obstacles (e.g. pedestrians, cars, bicycles) occluding the curbs.

As a result, some approaches reconstruct the curb structure in case of occlusion or noisy sensor data. (HERVIEU; SOHEILIAN, 2013) analyses the height variation analysis is performed by comparing the normal surface angle with a given threshold. The obtained points are placed in a bidimensional grid and then Kalman filter is employed to estimate the curb model. Similarly, (KELLNER et al., 2014) uses the iterative end point fit algorithm to obtain the curb candidates that is later stored in a grid. Each cell of the grid also contains curb features that are used to link nearby cells. Those segments associated with a high feature score are classified as curbs. (CHEN et al., 2015) employs Gaussian process regression to eliminate outliers from the curb

candidate points. These points are classified as curbs if their distance to the fitted line is lower than a threshold value.

Except for (CHEN et al., 2015), other works employ single layer laser sensor, which results in a solution that requires accumulating curb candidate points in a grid to get a dense measurement for a more robust result. The use of multilayer laser sensor dispenses this step and enables different approaches for curb detection.

The proposed curb detection method differs from above methods by using ring compression which is a surface slope information that is obtained in a straightforward way from multilayer laser sensor data (MONTEMERLO et al., 2008). Another advantage of our method is the use of robust regression to eliminate false-positives from the detection. Robust regression is a class of fitting methods that deal with noisy data and outlier rejection.

2.1.2 Road Marking Detection

Road markings (or horizontal signalizations) consist of the paintings situated on the asphalt, with the purpose to guide the drivers through the urban streets. Some examples of road markings are pedestrian crossing, lane marking and stop marking.

Although cameras have been widely used for road marking detection (HILLEL et al., 2012), camera images require supplementary processing to deal with illumination problems, such as shadow and light variation. As an alternative, laser sensor capable of returning infrared reflective intensity can be employed, with the advantage of suffering less external influence.

The method proposed by (KIBBEL; JUSTUS; FURSTENBERG, 2005) extracts the highest intensity points from the road surface for road marking segmentation. These points are then projected in a bidimensional grid and a sliding window runs in the lateral direction to build an intensity histogram. Values of the histogram associated with high gradient values are classified as lane marking. The method of (OGAWA; TAKAGI, 2006) follows a similar approach, but the Hough transform is applied on the projected points. (THUY; LEÓN, 2010) uses the intensity data of a single layer laser sensor is registered and stored in a grid. A sliding window passes through the grid and classifies window area as road marking if the number of high-intensity points exceeds a threshold. The main drawback of these methods is the sensibility to the rotation. Therefore, they are only suitable to lane marking detection.

Differently from other methods that use sliding window approach, (GUAN et al., 2015) employed a thresholding based method to segment road marking from the asphalt. Through thresholding method, it is possible to determine the best intensity value that separates road marking points from asphalt points. However, due to the presence of noises in the remittance intensity data, this method requires additional filtering.

The approach of (KAMMEL; PITZER, 2008) searches for contiguous segments of laser points with the same intensity. Just those points with intensity higher than the median are taken

into account. The detection is reinforced with Radon transform with the argument that it is more robust than the Hough transform.

The proposed method follows the approach of (GUAN et al., 2015), but employs Otsu thresholding. In order to mitigate the presence of noises in the remittance intensity data, the intensity values returned by the sensor are previously calibrated by a method presented by (LEVINSON et al., 2011). Another advantage of thresholding method is the possibility of detecting the majority of road marking and is not sensible to the rotation. Moreover, it is verified the contiguous intensity segments as (KAMMEL; PITZER, 2008) to guarantee a better detection.

2.2 Mapping

In map-aided localization, the quality of the map directly affects the vehicle localization performance. For an accurate localization, maps that get closer to the real environment are preferred.

Typically, metric maps are employed to reconstruct the environment. These maps are formed by geometric data and suited to capture fine details, but at cost of large memory storage, specially when representing wide areas (THRUN; BURGARD; FOX, 2005).

Occupancy grid map (OGM) is a popular metric map employed to represent urban structures through a binary grid. (QIN et al., 2012) use a high-resolution OGM is used to represent road curbs. Similarly, (HERVIEU; SOHEILIAN, 2013) employ OGM to accumulate the detected curb points and refined them through regression lines. Other works represent the road marking structures using OGM, such as (DEUSCH et al., 2014) and (KONRAD; NUSS; DIETMAYER, 2012). Some authors also use OGM to combine other urban structures. To exemplify, in (LEVINSON et al., 2011) and (SCHREIBER; KNÖPPEL; FRANKE, 2013) both curb and road marking (reflection intensity) data are stored in the OGM. These works employ grid with cell resolution between 0.10 m to 0.20 m to depict environment details. However, authors that employ the OGM for localization requires repetitive measurements in the same area or gather measurement when there is no occluding obstacle on the street (LEVINSON et al., 2011).

Aiming a precise map and at the same time avoiding the tedious process of gathering mapping data, authors recently have started to use geographic information system (GIS) maps provided from third party companies. In contrast to OGM, these maps are represented by vector graphics and submitted to fine manual adjustments. The work of (WEI; CAPPELLE; RUICHEK, 2013) use GIS maps to retrieve building facades. Differently, (VERONESE et al., 2015) make use of aerial maps to get access to road structures. High definition road markings in GIS format is also employed by (SUHR et al., 2016). The main drawback of GIS solution is the high cost to get access to these databases and is limited few locations.

In order to overcome the limitations of OGM and still generating a high-resolution map, the current work proposes the use of Gaussian process occupancy map (or continuous occupancy map). The main benefit of this structure is the possibility to infer the occupancy of any place, including unobserved areas, thus avoiding the necessity of repetitive measurements or using fine-grained grids (O'CALLAGHAN; RAMOS, 2012).

2.3 Localization

Localization methods based on maps estimate the vehicle position by associating the detected sensor features with the environment map. Usually, Monte Carlo localization (MCL) (a.k.a. particle filter localization) method is employed due to the facility to implement and robustness to noises (THRUN; BURGARD; FOX, 2005). In MCL, the vehicle position distribution is given by a set of particles. In this sense, authors employ different measurement likelihood functions to incorporate the sensor data in the particle set.

Some works employ Iterative Closest Point (ICP) to match the detected road features to the map. This approach is also known as map matching. The ICP is employed by (YONEDA et al., 2014) to match portions of the laser measurements in GIS maps. In a similar way, (SCHREIBER; KNÖPPEL; FRANKE, 2013) employ ICP to match lane markings to an OGM. (WEI; CAPPELLE; RUICHEK, 2013) use the distance of the detected building facades to a GIS map. The main drawback of this approach is the high processing time of ICP.

The most common approach is the use of likelihood field laser model which adds a Gaussian over the occupied area of the grid map. (QIN et al., 2012) employ the likelihood field laser model to obtain the likelihood of detected curb to correspond the OGM. The likelihood field values of all map can be computed beforehand to reduce the computing complexity. However, this likelihood model is less flexible in the sense that it allows the evaluation of a single feature.

Another popular likelihood function is the normal (or Gaussian) distribution probability density function which makes possible the incorporation of several features in a single model. In the work of (LEVINSON et al., 2011), this likelihood function is used to match detected curb and road markings to an urban map. Following this idea, (DEUSCH et al., 2014) proposed the use of probability density function to calculate the measurement likelihood that includes road marking and curb data. In the study of (SUHR et al., 2016), a set of detected lane marking features were combined through the same approach in order to calculate the vehicle position on a GIS map.

The use of normal distribution probability density function makes it possible to attach more than one feature in a single likelihood model. Considering the flexibility and robustness of normal distribution probability density function for localization, the proposed method employs it to formulate a likelihood function to support GPOM representation.

2.4 Final Remarks

This chapter presented a bibliography review of some components of map-aided localization system for autonomous vehicles: road feature detection, urban environment mapping and vehicle localization. Although earlier approaches have delivered satisfactory results, many of them were tested in favorable conditions. As a result, recent works started to explore the development of methods in challenging situations. For instance, (WEI; CAPPELLE; RUICHEK, 2013) developed a localization that works in heavy traffics or urban canyons, (DEUSCH et al., 2014) evaluated the localization method in rural roads and (SUHR et al., 2016) employed low-cost sensors for localization.

Following this idea, the proposed localization method aims at using maps built with incomplete sensor data for localization. The next chapters describe details of each component of the proposed localization system.

ROAD FEATURE DETECTION

Urban maps are formed by environment structures detected from sensor measurements. Here, we focused on the detection of two road structures to build the maps required for localization: curbs and road markings. These two traffic structures were chosen because of the facility to find them inside cities and the fact that they do not frequently change over the time. These structures also have been successfully employed in vehicle localization problems ([MONTEMERLO et al., 2008](#); [LUETTEL](#); [HIMMELSBACH](#); [WUENSCHKE, 2012](#); [HILLEL et al., 2012](#)).

The proposed curb and road marking detection relies on Velodyne HDL-32E multilayer laser sensor measurements. This sensor consists of a rotary column with several laser emitters that return in each revolution, a tridimensional point cloud formed by 32 concentric rings ([figure 1](#)). This sensor additionally returns infrared remittance intensity that makes possible analyze the reflectivity of the objects. The ring shape is employed to detect the road curbs, whereas laser infrared remittance intensity is taken into account to detect the road markings.

Details of the curb detection and road marking detection are presented in the next sections.

3.1 Curb Detection

The proposed curb detection is based on the obstacle detection method proposed by ([MONTEMERLO et al., 2008](#)). In this, obstacles are detected by comparing the distance between points belonging to adjacent rings to a threshold.

To describe the curb detection method, a scenario in which the multilayer sensor beams hit a flat surface is analyzed. In this situation, the sensor returns a set of concentric circles. The circle radius is inversely proportional to the angle formed between the beam and the surface, as shown in [figure 1](#). Thus, the radius of a ring is given by the following expression:

$$r_i^{\text{plane}} = \frac{h}{\tan \theta_i} = h \cot \theta_i, \quad (3.1)$$

where r_i^{plane} and θ_i correspond to the ring radius and the laser beam angle of the ring i , respectively. The sensor distance to the ground is denoted by h .

The distance between consecutive rings r_i and r_{i+1} is given by:

$$\Delta r_i^{\text{plane}} = r_{i+1}^{\text{plane}} - r_i^{\text{plane}} \quad (3.2)$$

$$= h \cot \theta_{i+1} - h \cot \theta_i \quad (3.3)$$

$$= h(\cot \theta_{i+1} - \cot \theta_i). \quad (3.4)$$

When the laser beams intercept an obstacle, these rings change the aspect presented in [figure 1](#) by losing the circular shape and start to compress nearby the obstacle. Therefore, when the distance between rings gets lower than $\Delta r_i^{\text{plane}}$, it means that an obstacle has been detected.

Figure 1 – Representation of concentric rings returned by the multilayer laser sensor when the beams hit a flat surface.

Figure 2 – Rings returned from laser measurement compress with rates proportional to surface slope. In this laser frame, different obstacles can be visually distinguished, such as a wall (yellow area), grass (green area), street (blue area) and curbs (red area).

Analyzing a point cloud returned by a single frame of a laser sensor, a different compression pattern is observed for a particular obstacle shape. For instance, in [figure 2](#) the wall (yellow area) produces higher compressed rings, vegetation (green area) results oscillated compression, street (blue area) produces low compressed rings, and curbs (red area) are associated with ring compression rate between street and wall. In this sense, an obstacle may be identified by verifying the ring compression rate that it produces.

In other words, curbs may be detected by checking if the distance between points of adjacent rings is inside a given interval. This interval is defined as:

$$S_i = \left[\alpha \Delta r_i^{\text{plane}}, \beta \Delta r_i^{\text{plane}} \right], \quad (3.5)$$

where S_i is the valid interval for the ring r_i , α and β control the interval bounds, with $\beta > \alpha$. This interval gets larger towards to the external ring, due to the increase of the space between rings.

To simplify the distance calculation of adjacent ring points, each sensor measurement is projected into a circular grid with a fixed cell size as shown in [figure 3](#). Each circular row stores points of a single ring. The circular grid cell is denoted as $c_{i,j}$, where i corresponds to the ring index and j to the angular index. Cells store the average value of the points that fall in. Specifically, the average point position (x, y, z) and the Euclidean distance d to the sensor relative to the ground plane (xy) . Points higher than 0.30 m (maximum tolerance for curb height) are automatically discarded to avoid storage of unnecessary data.

Through circular grid, ring compression is given by the difference between euclidean distance of adjacent cells:

$$\Delta d_{i,j} = |d_{i,j} - d_{i+1,k}|,$$

where $\Delta d_{i,j}$ denotes the compression value of cell $c_{i,j}$, and $d_{i,j}$ is the euclidean distance of $c_{i,j}$ to the grid center (sensor position). The $c_{i+1,k}$ is the closest cell to $c_{i,j}$ in radial direction.

Therefore, cells with $\Delta d_{i,j}$ lower than a given threshold have higher chances to belong to a curb structure. The calculation of this threshold is covered later in this chapter.

Figure 3 – Circular grid representation used to manage laser point cloud during curb detection. Each circular row stores point cloud belonging to a single ring. Cells store the average values of those points that fall in.

The exclusive use of ring compression method is susceptible to false-positives, because objects with a similar slope as curbs may be incorrectly detected. To improve the curb detection, filters were associated with this method. The first one was the gradient filter which performs a complementary analysis of the surface slope and the second one was the occlusion filter that makes the method insensitive to objects that cause occlusion to the curbs (e.g. few cars parked beside the curbs). Details of these filters are presented in the next subsections.

3.1.1 Gradient Filter

In contrast to ring compression which performs a comparison between cells of adjacent rings, the gradient filter analyzes cells belonging to the same ring. Specifically, for a given cell, the height variation between its predecessor and successor cells of the same circular row are taken into account to calculate the surface slope (i.e. gradient).

The gradient $\nabla g_{i,j}$ of cell $c_{i,j}$ is then computed through the following expression:

$$\nabla g_{i,j} = \frac{z_{i,j+1} - z_{i,j-1}}{\text{dist}(c_{i,j+1}, c_{i,j-1})}, \quad (3.6)$$

where $z_{i,j}$ is the height value of a cell $c_{i,j}$ and $\text{dist}(a,b)$ is the Euclidean distance function between points a and b in the (x,y) Cartesian coordinate reference.

For each cell $c_{i,j}$, we compute the $\nabla g_{i,j}$ and compare with a t_s threshold to classify it as a curb candidate or not.

3.1.2 Occlusion Removal

Obstacles present on the streets may have different slopes and geometries that make ring compression with gradient filter still insufficient to remove the false positives. Furthermore, these obstacles may occlude the laser readings and then reduce the field of view of curbs. This situation is depicted in [figure 4a](#) in which a car blocks the sight to the left curb.

To remove occluding obstacles, a regression method robust to outliers is applied to fit a curve over the curb candidates. Therefore, the fitted curve separates the curb points from the noisy points. Precisely, a method of robust regression named least trimmed squares (LTS) is employed. In comparison to other fitting methods, LTS computes the function parameters considering the presence of outliers. [Figure 4b](#) shows the curve model of road data estimated by three different fitting methods and [table 1](#) provides their performance. Although the spline fitting obtained the best sum of squared errors of prediction (SSE) and the coefficient of determination (R^2), it resulted in the shortest distance to the outliers. The performance of the LTS regression was slightly lower than that of the spline. However, satisfactory curve models were obtained in the presence of outliers.

The LTS method is based on the robust regression theory. One of its advantages is that it computes the function parameters of the data subset that minimizes the cumulative error (i.e.

	SSE	R ²	Dist. to Outlier
Least squares (quad. pol.)	27.68531	0.59519	1.33402
Spline (p = 0.99)	0.48798	0.99286	0.07516
LTS (quad. pol.)	1.33002	0.97789	2.23326

Table 1 – Comparison of the performance of different fitting methods for the estimation of the curb model.

Figure 4 – Curb detection in an occluded road. (a) Occlusion caused by a car on the street. False-positives (yellow points) are removed by estimating the road shape (blue line) from other detected points. (b) Performance comparison of curve fitting methods. Axes are in meter unit.

the subset that does not contain outliers) (ROUSSEEUW; DRIESSEN, 2006). In the LTS fitting, considering a dataset with n points and a subset with size h without outliers ($\frac{n}{2} < h < n$), the function parameters are obtained by minimizing the following function:

$$f_{lts}(\gamma) = \arg \min_{\gamma} \sum_{i=1}^h \varepsilon_i^2(\gamma), \quad (3.7)$$

where $\varepsilon(\gamma)$ is the residual function and γ is the curve function parameters. The quadratic polynomial was chosen as the basis function for LTS fitting.

The number of combinations in LTS fitting is given by $\frac{n!}{h!(n-h)!}$. When n is relative low, the exact solution is calculated. As n increases, the number of combinations exponentially grows, thereby an approximate solution is employed.

The curve parameters are computed separately for left and right sides of the road. After the curve parameters have been obtained, only those curb candidate points that have a distance to the fitted curve $f_{lts}(\gamma)$ smaller than a threshold t_d are preserved.

Figure 5 shows the effect of each curb detection step to get the final result. This scenario depicts a steep road with a car occluding the left curb. In figure 5a the ring compression analysis is prone to false-positives in a non-flat terrain. The gradient filter eliminates laser points with a slope that do not correspond to curb structures (figure 5b) and regression filter eliminates laser points originated by occlusion through LTS fitting (figure 5c).

Figure 5 – Curb detection steps in an occluded scene in a steep street. This scenario illustrates the effect of gradient and regression filter to remove false-positives generated by ring compression.

3.1.3 Parameter Tuning

We can manually test several values for parameter set $\Theta = \{\alpha, \beta, t_s, t_d\}$, $\Theta \in \mathbb{R}$ to obtain satisfactory results for curb detection. As this approach is impractical and the best parameters may not be achieved, we automatized the parameter search through a tool named Meta-Evolver¹, in which an objective function must be defined for problem minimization. The following equation was adopted:

$$f_{\min}(\Theta) = \arg \min_{\Theta} \varepsilon(\Theta), \quad (3.8)$$

where $\varepsilon(\Theta)$ is the mean squared error (MSE) of the estimated curb model using parameters Θ in relation to the ground truth (actual curb model). The function $\varepsilon(\Theta)$ is defined as:

$$\varepsilon(\Theta) = \frac{1}{n} \sum_{i=1}^n (\mathbf{k}_i(\Theta) - x_i)^2, \quad (3.9)$$

where $\mathbf{k}_i(\Theta)$ refers to the position of the i -th detected curb point using parameter set Θ , x_i is the curb point position ground truth and n is the total amount of points classified as a curb. A situation is considered optimal when a low MSE is obtained.

The first step of Meta-Evolver is to take a random walk in the parameter space based on arbitrary values of Θ . After obtaining few values of Θ and $\varepsilon(\Theta)$, a linear function is fitted. The direction in which Θ values must be explored is given by the resulting linear function. The search is stopped after reaching the maximum number of iterations or no more changes are observed in Θ .

3.2 Road Marking Detection

The multilayer laser sensor adopted in this work returns infrared reflectivity intensity together with range distance measurement. The reflectivity intensity ranges from 0 to 255. This

¹Meta-Evolver: <http://sourceforge.net/projects/annevolve>

intensity is related to the energy necessary to a beam return after hitting a surface. For shiny surfaces, such as road markings, low intensities are returned and for rough surfaces, such as asphalt, high intensities are returned. In this way, the intensity contrast of point cloud belonging to road surface is analyzed to detect road markings.

A paved road with horizontal signalization has the predominance of high-intensity values rather than low-intensity values. This scenario can be depicted as a grayscale image with the asphalt as the background and the road marking as the foreground (see figure 6a). The intensity histogram of a single ring that passed through a region with both asphalt and road marking is illustrated in figure 6b. Two peaks are present in this histogram which represent the background and foreground classes. Moreover, the intensity value between these two peaks determines the boundary of these two classes. Therefore, the points of the ring with reflectivity values higher than this intensity are classified as road markings. In image processing field, thresholding is a popular method that uses this same principle to segment foreground from background.

Here, a method based on Otsu thresholding (OTSU, 1979) is employed for road marking detection. The original Otsu algorithm receives a grayscale image as input and calculates the threshold T that maximizes the variance between two classes (or modes) of the image's intensity histogram. In this way, a segmented binary image I_B is obtained by applying the following comparison in the original image I_o :

$$I_B(x,y) = \begin{cases} \begin{matrix} F0 \\ F1 \\ F2 \\ F3 \end{matrix} & 1, \text{ if } I_o(x,y) > T \\ \begin{matrix} B0 \\ B1 \\ B2 \\ B3 \end{matrix} & 0, \text{ if } I_o(x,y) \leq T \end{cases}$$

Despite Otsu thresholding has been designed for image inputs, it is possible to work with intensity point cloud because the main processing is performed on an intensity histogram. In this sense, Otsu method was adapted to support point cloud data.

Considering that Otsu method has better results when the point cloud intensities present a

Figure 6 – Road marking detection based on infrared intensity analysis. (a) Raw intensity data returned by the laser sensor artificially colored according to the intensity value. Note that there are a substantial difference between road marking and asphalt points. (b) Intensity histogram of points belonging to a single ring represented in the left image. Two classes are present, grouped by the peaks of the function in red.

bimodal distribution, one must ensure that only measurements that hit the street will be processed. If all the surrounding data were used, the histogram may not show a bimodal shape due to the possibility to have several objects with varied reflectivities in the scene as shows [figure 6a](#). For this reason, the road curb curves (left and right) obtained from curb detection are used as boundaries to get only those points situated on the street (details in [section 3.1](#)). Therefore, only points that are between these two curves are fed to Otsu method. In addition, sensor intensity values must be calibrated in a previous step to avoid incorrect measurements. For laser sensor intensity calibration, the deterministic calibration proposed by Jesse Levinson was employed ([LEVINSON et al., 2011](#)).

After obtaining the points from street surface, Otsu method is applied separately in each ring of the incoming point cloud. This is necessary due to each emitter of the sensor returning a different intensity for the same surface. Despite the sensor calibration step can mitigate this difference, it is not guaranteed that all emitters have the same behavior. The calculation of threshold T of each ring comprises the following steps:

1. Normalize the intensity histogram obtained from points of a single ring. Let n_i the i -th value of the histogram, M the total amount of points of the histogram, the normalized value p_i is given as:

$$p_i = \frac{n_i}{M}.$$

2. Calculate the cumulative sum $P(k)$ of the normalized histogram, with $0 < k < L - 1$ and L the intensity range that a point can represent (in this case, $L = 256$). The cumulative sum represents the probability of a point to belong to a class that covers the range $[0; k]$ (i.e. the class that represents road markings). The cumulative sum is calculated as follows:

$$P(k) = \sum_{i=0}^k p_i.$$

3. Calculate the cumulative mean $m(k)$ which represents the mean intensity of the class that is inside the interval $[0; k]$, through the following summation:

$$m(k) = \sum_{i=0}^k i p_i.$$

4. Calculate the global cumulative sum m_G which gives the mean of the overall histogram:

$$m_G = m(L - 1) = \sum_{i=0}^{L-1} i p_i.$$

5. Calculate the global variance σ_G^2 to obtain the sum of the variance of all points of the ring:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i.$$

6. Calculate the variance between classes $\sigma_B^2(k)$ by the expression below. The value k that maximizes this variance is assigned to threshold T which later will be used to separate the histogram classes.

$$\sigma_B^2(k) = \frac{(m_G P(k) - m(k))^2}{P(k)(1 - P(k))},$$

$$T = \arg \max_{0 \leq k \leq L-1} \sigma_B^2(k).$$

7. Calculate the separability between classes denoted by $\eta(T)$. This measure returns values between 0 and 1 that indicate the performance of T to separate the histogram into two classes. Therefore, we discard the classification when T results in $\eta(T) \leq t_\eta$, where t_η is the separability threshold. The separability is calculated as follows:

$$\eta(T) = \frac{\sigma_B^2(T)}{\sigma_G^2}.$$

8. Check if $T < t_T$, where t_T defines the maximum allowed value of T . This is necessary because road markings are associated to low intensities, thereby, in some cases, the presence of noise on the asphalt without road markings may result in $\eta(T) > t_\eta$.
9. Finally, classify points of the ring with intensity lower than T as road marking candidates.

Because of the sensor noise and the presence of unexpected objects on the street, few points still may be incorrectly detected as road marking candidates. Considering that horizontal signalization have a smaller width compared to the road width, false-positives are removed by analyzing the length of the contiguous sequence of road marking candidate points. This approach is based on the work of (KAMMEL; PITZER, 2008) that remove road marking candidate segments that are oversized. Consequently, a set of contiguous points classified as road marking with the length larger than a predefined threshold is automatically rejected.

3.3 Final Remarks

This chapter described the proposed road feature detection methods. As curbs are more frequent than road markings, the first structure is taken as the main environment feature and the second structure is used as an auxiliary information in the localization system. Moreover, curbs also provide the free area between the left and right curbs. In this sense, representing street curbs through Gaussian process occupancy maps may lead to a rich representation of the environment.

The following chapter introduces occupancy grid mapping and Gaussian process occupancy mapping methods along with the adaptations made to enable the representation of large areas.

ENVIRONMENT MAPPING

Metric maps are preferred to topological maps especially because of the possibility to represent finer details of the environment and deliver higher accuracy during localization (FOX et al., 2001). Occupancy grid map (OGM) is a well known metric map that is widely adopted in the robotics field. Gaussian process occupancy map (GPOM) is a novel metric representation that overcomes the OGM main limitations: environment discretization and assumption of independence between sensor measurements.

This chapter briefly covers the occupancy grid mapping and then describes the Gaussian process theory along with Gaussian process occupancy mapping. For simplicity, here we assume that the robot position is known during whole mapping process.

4.1 Occupancy Grid Mapping

Among the structures used to store and represent the environment information, occupancy grid map (OGM) is the most popular in virtue of the relative easiness to implement and its high precision (THRUN et al., 2004).

OGM represents the environment by a fine-grained evenly spaced bidimensional grid in which the cell informs the occupancy probability of the area that it covers. Formally, the occupancy of a map cell at index i is denoted as $p(m_i)$, where i corresponds to the linear index of a bidimensional grid m .

The mapping process consists in calculating the map posterior from the sensor measurement and robot poses. However, as the number of map states is numerically intractable, the independence between cells is assumed to calculate the posterior probability of the map. As a consequence, given the sensor measurement and the robot position, the map posterior is calculated as follows:

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t}), \quad (4.1)$$

where $z_{1:t}$, $x_{1:t}$ and t correspond to sensor measurement, robot position and time stamp, respectively.

The cell occupancy probability is obtained by applying Bayes filter:

$$p(m_i|z_{1:t}, x_{1:t}) = \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} \frac{p(m_i|z_{1:t-1}, x_{1:t-1})}{1 - p(m_i|z_{1:t-1}, x_{1:t-1})} \frac{1 - p(m_i)}{p(m_i)}, \quad (4.2)$$

where $p(m_i)$ is the occupancy prior of m_i .

As the occupancy probability calculation involves subsequent multiplications of values between 0 and 1, this can produce infinitesimal values that lead to truncation problems (THRUN; BURGARD; FOX, 2005). For a better computational tractability, log odds formulation is employed:

$$l(x) = \log \frac{p(x)}{1 - p(x)} \quad (4.3)$$

Consequently, the occupancy probability of cell c_i is rewritten as follows:

$$p(m_i|z_{1:t}, x_{1:t}) = 1 - \frac{1}{\exp(l_{i,t})}, \quad (4.4)$$

where $l_{i,t}$ is the cell occupancy in the log odds form:

$$l_{i,t} = \sum_{k=1}^{t-1} \log \frac{p(m_i|z_{1:k}, x_{1:k})}{1 - p(m_i|z_{1:k}, x_{1:k})} - (t-1) \log \frac{p(m_i)}{1 - p(m_i)}. \quad (4.5)$$

4.2 Gaussian Process

The GPOM is based on a supervised machine learning method named Gaussian Process (GP). The central idea of GP is to explore the relationship between input and its corresponding target values through a given kernel (covariance function). GP is generally employed in regression problems with the advantage of having a fast learning process and is tolerant to noisy training dataset (SEEGER, 2004). Differently from other regression methods, GP does not require the prior definition of the basis function of the curve that models the dataset. In other words, it is not necessary to previously know the shape of the curve (e.g. linear, quadratic, cubic) before applying GP regression. This property makes the GP a non-parametric method, thus fewer parameters are needed to adjust during the training step.

A GP is a multivariate Gaussian distribution modeling the function space of a dataset $\mathbb{D} = \{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. The model is fully specified by a mean function $\mu(x)$, and a covariance function $k(x, x')$. The covariance function is also known as kernel.

The GP prior of a query data x^* (i.e. the test value) is a Gaussian distribution with mean at 0 and variance given by the covariance of the query point as follows:

$$y^* = \mathbb{P}(0, k(x^*, x^*)). \quad (4.6)$$

Given the GP prior and conditioning on the training dataset $\mathcal{D}$, the posterior probability y^* is also a Gaussian:

$$p(y^* | \mathcal{D}, x^*) = \mathcal{N}(\mu(x^*), \sigma(x^*)). \quad (4.7)$$

The mean and the variance of the posterior are represented respectively by $\mu(x^*)$ and $\sigma(x^*)$, which are obtained by:

$$\mu(x^*) = k(x^*, x)^\top [k(x, x) + \sigma_n^2 \mathbf{I}]^{-1} y, \quad (4.8)$$

$$\sigma(x^*) = k(x^*, x^*) - k(x^*, x) [k(x, x) + \sigma_n^2 \mathbf{I}]^{-1} k(x, x^*), \quad (4.9)$$

where σ_n is the global noise value of the dataset $\mathcal{D}$. The Gaussian Process of x^* is also denoted as $\mathcal{GP}(\mu(x^*), \sigma(x^*))$ ¹.

The calculation of the posterior makes the GP distribution have a higher confidence about the function space by including observation points (i.e. training data). In this sense, the GP model relies on a representative training dataset. Figure 7 illustrates the addition of observation points into the GP prior to generate the GP posterior. The uncertainty (denoted by the gray area) gets thinner and the space of functions starts taking shape.

Among the covariance functions, squared exponential is a common choice for general problems. This covariance is defined as the follows:

$$k(x, x') = \sigma_f^2 \exp \left(-\frac{(x - x')^2}{2l^2} \right), \quad (4.10)$$

where x and x' are covariance input values, l the characteristic length scale and σ_f^2 the maximum value of the function.

Figure 7 – Functions randomly sampled from the GP distribution and the corresponding uncertainty (gray area). (a) Before the inclusion of the training dataset, the uncertainty is large and consequently the space of function is wide. (b) After adding training samples (plus symbol), the uncertainty reduces, in special around the training points. The space of functions starts to get a delineated shape. Figures extracted from (RASMUSSEN; WILLIAMS, 2005).

¹The full mathematical derivation is presented in the chapter 2 of (RASMUSSEN; WILLIAMS, 2005).

Figure 8 – Effects of different parameters l on the GP regression. Curve smoothness and the confidence changes significantly. Blue points, plus symbol and gray area denotes query values, training data and variance, respectively. Figures extracted from (RASMUSSEN; WILLIAMS, 2005).

Typically, when x and x' are close points, the covariance tends to values near σ_f^2 and when x and x' are far each other, the covariance returns values next to 0. The way that this proximity influences on how fast or slow the covariance goes towards to 0 or σ_f^2 is controlled by l . In figure 8 is shown the impact of the parameter l on the GP posterior. Note that subtle variation in l may bring drastic changes in the regression.

Though the chosen parameters for the covariance function may interfere in the regression outcome, GP has tools to avoid such issue. The term hyperparameter is widely used to refer to the GP parameters, here denoted by $\theta = \{\sigma_f, \sigma_n, l\}$. To learn the best values of hyperparameter for both mean and covariance functions, the marginal likelihood of the data is maximized through the log marginal likelihood function (RASMUSSEN; WILLIAMS, 2005):

$$\log p(y|x, \theta) = \frac{1}{2} y^T K^{-1} y - \frac{1}{2} \log |K| - \frac{n}{2} \log 2\pi, \quad (4.11)$$

where K is the covariance matrix of the training dataset $\frac{m}{n}$ with size n . In a practical approach, these parameters are obtained through optimization methods such as gradient descent and simulated annealing.

4.3 Gaussian Process Occupancy Mapping

Some adjustments are necessary when applying GP regression for robot mapping. The following subsections describe the transition from GP theory (presented previously) to Gaussian Process Occupancy Map method.

4.3.1 Point and Line Observations

Changing the mean and variance formulations of equation 4.8 and equation 4.9 to the mapping context, x_i corresponds to the laser beam positions and $y_i \in [0, 1]$ to the occupancy values of x_i , specifically, 0 for free and 1 for occupied location. A single laser beam is geometrically equivalent to a line with a starting and ending point. The end point is where the beam hit an

object and the extent between start and end points is the space free of obstacles. Therefore, two types of observation arise from a single beam with a valid measurement: a point representing the occupied area and a line representing the free area. Hence, the values of dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$ are expressed as $x_i \in \{x; l(u, v) | x, u, v \in \mathbb{R}^2\}$, where x denotes a point observation and $l(u, v)$ a line observation with u and v as the line boundary points.

4.3.2 Covariance for Line Observations

As covariances were originally formulated to work with point values (e.g. equation 4.10), these functions must be adapted to work with line observations. Two solutions were proposed by (O'CALLAGHAN; RAMOS; DURRANT-WHYTE, 2010): (a) split the line into evenly spaced points; (b) use integral kernels. The first one sets a fixed interval to discretize the lines into a set of points. Despite a straightforward approach, this solution may require a fine-grained discretization which can lead to a substantial increase of the observation dataset $\mathcal{D}$. The second one employs path integral to allowing covariance calculation for line entities. This preserves the continuous nature of the line, though generating a more smooth result that blurs the space nearby the lines.

In figure 9 is illustrated the GPOM generated from a line and a point observation using the aforementioned approaches. Note that the approach (a) generates a less smooth transition from free to occupied region compared to approach (b).

The next subsection describes the integral kernel method for line covariance calculation.

4.3.2.1 Integral Kernel

Usually, we use points to represent the training dataset to build the GP model. In the GP mapping problem, the laser beam lines could be discretized into several points to represent the free space region, however, this solution may cause loss of information or depending on the sampling size, slow down the GP performance, due to its $\mathcal{O}(n^3)$ complexity.

Figure 9 – Representation of a single laser scan using discrete (OGM) and continuous (GPOM) representations. The training (observation) data set was overlapped in the GPOM images (yellow corresponds to free space and green to occupied space). Blue and red colors are associated with free and occupied regions, respectively. Axes scale is in meters.

To preserve the line information as a single continuous observation, integral kernels are introduced to allow the covariance computation between other geometric entities such as lines and surfaces. Essentially, the integral kernel makes use of path integral theory to redefine the covariance function. The covariance between a curve and point through integral kernel is defined as:

$$k(l, x) = \int_a^b k(l(u), x) du, \quad (4.12)$$

where, a and b correspond to the start and end point of the curve represented by the function $l(u)$, and x is a point. Then, the covariance between two curves is given by the following equation:

$$k(l_1, l_2) = \int_a^b \int_c^d k(l_1(u), l_2(v)) dudv, \quad (4.13)$$

where c and d denote the start and end point of the second curve $l_2(v)$. In the case of GPOM, l_1 and l_2 are two dimensional line functions. Taking two points $a = (x_s, y_s)$ and $b = (x_e, y_e)$ which represent the line start and end points, thus, the line function is defined as $l(u) = (x(u), y(u)) = (x_s + u(x_e - x_s), y_s + u(y_e - y_s))$, where $u \in [0; 1]$.

Considering the absence of analytical solutions for the covariance between lines (Eq. 4.13) (O'CALLAGHAN; RAMOS, 2011), numerical solution of the integral using Clenshaw-Curtis quadrature is used (WALDVOGEL, 2006). Thus, the integral is rewritten as:

$$\int_l f(x) dx \approx \sum_{k=0}^n w_k f(x_k), \quad (4.14)$$

where, $x_k = \cos v_k$ with $v_k = k\frac{\pi}{n}$, $k \in [0, n]$ and n is the number of samples (quadrature order).

The Clenshaw-Curtis weights w_k are obtained by:

$$w_k = \frac{2}{n} \left(1 - \sum_{j=1}^{\lfloor n/2 \rfloor} \frac{b_j}{4j^2 - 1} \cos(2jv_k) \right), \quad (4.15)$$

where,

$$b_j = \begin{cases} 1, & j = n/2 \\ 2, & j < n/2. \end{cases} \quad (4.16)$$

In this work we used the Matérn family of covariance functions (RASMUSSEN; WILLIAMS, 2005) which is defined as:

$$k(x, x') = \sigma^2 \frac{1}{\Gamma(\nu) 2^{2\nu-1}} \sqrt{\frac{2\nu}{(x-x')^2}} K_\nu \sqrt{\frac{2\nu}{(x-x')^2}}, \quad (4.17)$$

where, ν and l are positive parameters and K_ν is a modified Bessel function. Specifically, the Matérn covariance function defined with $\nu = 3/2$ and the squared exponential covariance (which is a special case of Matérn function when $\nu \rightarrow \infty$) were considered in our experiments.

Figure 10 – GP regression using point and line observations. Estimated mean and variance are represented by red dot and the colored area.

4.3.3 Occupancy Prediction for New Point Observations

The GP was originally proposed to solve regression problems, therefore its original formulation must be adapted for classification problems, as in the case of GPOM. In this sense, the classification is used to restrict the mean result of GP inference to $[0, 1]$ interval. Basically, a squashing function is applied to the mean result for the achievement of the classified result. Here, the probabilistic least squares function (RASMUSSEN; WILLIAMS, 2005) is used:

$$p(\text{occupancy}|x^*) = \Phi \left(\frac{\alpha \mu(x^*) + \beta}{1 + \alpha^2 \sigma(x^*)^2} \right), \quad (4.18)$$

where Φ is the cumulative Gaussian distribution and the parameters α and β are obtained by leave-one-out cross-validation. Note that both mean and variance values obtained from the GP regression are used in the calculation of the occupancy value.

4.4 GP Regression Example

In figure 10 is illustrated an example of GP regression using point and line observations. The line observations represent the mean target value of the input values and the points relate one single input with a target. The estimated mean value of the regression function is denoted by red dots and the variance represented by the shaded area.

We can notice that the variance is wider in regions where there are few or none observation and get thinner in regions with some observation. In this way, the certainty of a certain location is directly related to prior information that we have.

4.5 Gaussian Process Occupancy Mapping

As stated before, the mapping process uses the GP classifier to determine the occupancy of a given point in a continuous space, named here as test point or test data. For this, sensor readings (e.g. laser scans) collected in a previous stage are used as training data to infer the occupancy of these points. Consequently, as opposite to the discrete mapping methods (e.g. occupancy grid maps), the GP can generate continuous maps and even determine the occupancy of locations which were not observed before. The continuity can be verified in [figure 9](#) which maps a single scan through OGM and GPOM. We can notice that GPOM generates a more dense occupancy map compared to OGM. As a consequence, a sparse laser sensor can simulate a high-resolution laser sensor when used GPOM.

Despite generating a more accurate map, few drawbacks arise when used GPs. First of all, considering a training data of size n , the complexity of the GP inference is $\mathcal{O}(n^3)$. This complexity is associated with the matrix inversion. Second, it requires $\mathcal{O}(n^2)$ of storage for inference.

Moreover, because GP is a kernel method, all training data is loaded at once during the inference process. As a consequence, the memory consumption demanded by GP mapping increases considerably according to the environment size. This even aggravates when performing matrix operations such as covariance calculation and matrix inversion. In this sense, modifications in GP mapping were introduced to enable representation of large environments. The next subsections describe some adaptations made in GP mapping method.

4.5.1 Updating Inverse Matrix Through Cholesky Factor

To avoid recalculations of covariance matrix inverse for every incoming observation set, the GP can be modified to calculate the inverse once and update the inverse matrix by Cholesky factor update ([OSBORNE et al., 2008](#)). In this way, given a dataset K_1 with covariance matrix $\begin{bmatrix} K_{1,1} & K_{1,2} \\ K_{2,1} & K_{2,2} \end{bmatrix}$ and Cholesky decomposition $\begin{bmatrix} S_{1,1} & S_{1,2} \\ 0 & S_{2,2} \end{bmatrix}$, the inclusion of dataset K_2 will lead to a covariance matrix $\begin{bmatrix} K_{1,1} & K_{1,2} \\ K_{2,1} & K_{2,2} \end{bmatrix}$ and Cholesky decomposition $\begin{bmatrix} S_{1,1} & S_{1,2} \\ 0 & S_{2,2} \end{bmatrix}$. Each block of the updated Cholesky is obtained as follows:

$$S_{1,1} = R_{1,1} \quad (4.19)$$

$$S_{1,2} = R_{1,1}^T K_{1,2} \quad (4.20)$$

$$S_{2,2} = \text{cholesky}(K_{2,2} - S_{1,2}^T S_{1,2}) \quad (4.21)$$

With Cholesky update, the matrix inversion complexity is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. Despite a considerable reduction, the $\mathcal{O}(n^2)$ complexity still makes necessary further optimizations when considering large inputs. In this sense, the way the data is handled also can be modified as described in the following subsections.

4.5.2 Information Theoretic Compression of Training Data

When gathering sensor data, redundant measurements are inevitably collected which could be eliminated for computing speed up. The unnecessary data can be discarded by adapting the information-theoretic compression of laser data proposed by (STACHNISS; KRETZSCHMAR, 2017). The idea behind the compression is to evaluate the mutual information of a measurement and the current dataset. In other words, only those readings that reduce the uncertainty about the environment are stored. Formally, the mutual information of the measurement set Z and the map m which represents the environment is given by:

$$I(m; Z) = H(m) - H(m|Z), \quad (4.22)$$

where I denotes the mutual information and H the entropy. Here m is an occupancy grid map which assumes independence between measurements.

To reduce the uncertainty, we must obtain the subset Z^* of Z in which:

$$Z^* = \underset{Z \subseteq Z_{1..n}, |Z| \leq n}{\operatorname{argmax}} H(m) - H(m|Z), \quad (4.23)$$

where n is the number of measurements.

The conditional entropy $H(m|Z)$ is given by the mean value of the entropy of each single measurement z from Z :

$$H(m|Z) = \int_Z p(z) H(m|z) dz. \quad (4.24)$$

From this, the conditional entropy of a single measurement can be calculated by:

$$H(m|z) = \sum_{C \in m} H(C|z) = - \sum_{C \in m} \sum_{c \in C} p(c|z) \log p(c|z), \quad (4.25)$$

where C denotes a subset of cells from map m and c a single cell from map m . The value of the occupancy $p(c|z)$ can be retrieved through the inverse sensor model method.

In order to simplify the compression, we calculate $H(m|z)$ of all measurements Z and remove those that result in a conditional entropy lower than the average $H(m|Z)$. Here we

Figure 11 – Example of information theory data compression applied in a laser dataset. Red segments represent the unnecessary laser beams removed from the original dataset. In this example, 1789 of 2941 beams were discarded (60.38% reduction), though visually the dataset reduction is practically unnoticeable.

consider each laser beam as a single measurement z . Therefore, the resulting subset Z^* is formed by measurements that satisfy the following condition:

$$Z^* = \{z \in Z \mid H(m|z) \leq H(m|Z)\} . \quad (4.26)$$

To exemplify, [figure 11](#) illustrates the compression of a laser dataset comprised by 2941 beams. After the mutual information analysis, the dataset was reduced 60.38% and almost visually imperceptible.

4.5.3 Mixture of Gaussian Processes

Instead of generating one single map, it is possible to split the training dataset into smaller regions and then produce a set of independent GPOMs. The strategy of using several GPs is named mixture of GPs and its application for mapping was proposed by ([KIM; KIM, 2012](#)).

The first step of the mixture of GPOMs is to cluster the training dataset according to a certain criteria. In the current work, k-means clustering method was applied. Given the maximum number of laser data s that each GP must handle and the training dataset with size n , the maximum number of clusters k is set as follows:

$$k \geq \frac{n}{s} . \quad (4.27)$$

After clustering, each training subset $\{d_i\}_{i=1}^k$ is associated with its corresponding centroid $\{c_i\}_{i=1}^k$. During the GP inference, the training subset z_i will produce its own model called GP expert, referred here as ε_i .

When building the environment map, a set of test points $D = \{p_j\}_{j=1}^N$ must be evaluated, with N the total number of test points. A gating network evaluates which expert should be chosen to infer the occupancy of the test points. For a test point p_j , we associate the expert ε_i whose corresponding cluster c_i is the closest to this point.

Figure 12 – Grouping subsets of training dataset using k-means. The clusters are bounded by rectangles and the centroid represented by the black star.

It is important to not assign a high value for k because it can result in clusters with low amount of training data which compromises the quality of the inference result by the GP expert. Figure 12 illustrates the dataset clustering using different values for k . Note that $k = 30$ results in considerably small clusters.

4.5.4 Combining Heuristics for GPOM

Aforementioned strategies to optimize the GPOM (Cholesky update, training dataset compression and mixture of GPs) can be combined to result in a better inference performance than if used a single one. This work incorporates these heuristics during the mapping process.

In the first step, the training (observation) dataset is reduced by applying the information theoretic compression. The remaining data are grouped through k -means and each cluster is associated with a GP expert. For each incoming training data, the Cholesky factorization is employed to update the inverse matrix.

After associating each training data subset with the corresponding GP expert, the network gateway selects the closest GP expert to a given test point. The occupancy resulted from the GP expert inference is stored in the GPOM. After processing all test points, the complete representation of the environment is generated. In figure 13 is represented the workflow to process training dataset and test dataset. Note that the GP experts are obtained in a totally separated step through offline processing.

Figure 13 – Heuristics to enable GP mapping of large areas. (a) workflow to associate the training data to the clusters and (b) gating network to set the GP expert to infer the occupancy of each test point.

4.5.5 Mathematical Optimization

One of the main bottlenecks of the GP is the covariance calculation. Besides that, depending on the data entity (e.g. point, line or plane) the complexity can increase substantially.

In the case of GPOM, covariance calculation of lines has much higher complexity than points due to the use of path integrals. Essentially, path integrals are used to approximate the analytic solutions of covariance calculations.

Particularly, the covariance between lines for the squared exponential kernel is known, but the Matérn kernel is more suitable due to its smoothness. The analytic solution for the line covariance using Matérn 3/2 kernel was attempted, but not successfully concluded. However, this strategy is still been under study for performance improvement.

4.6 Final Remarks

This chapter presented the occupancy grid mapping method that is widely adopted for localization problems and Gaussian process occupancy mapping method that will be employed in the proposed vehicle localization. The next chapter describes the Monte Carlo localization (MCL) method which is typically used together with OGM. Adaptations in the MCL algorithm to make suitable to GPOM is also described.

MONTE CARLO LOCALIZATION

Among the map-aided localization methods, Monte Carlo localization (MCL) is one of the most adopted due to its superior robustness and facility to implement. The main characteristic of MCL is the capacity to keep several hypotheses of the position. This makes the method more tolerant to sensor noise if compared to other localization methods, such as Kalman filter and Markov localization (FOX et al., 2001).

This chapter describes the basic MCL algorithm and the necessary adaptations to support occupancy grid map (OGM) and Gaussian process occupancy map (GPOM).

5.1 General Algorithm

Monte Carlo localization (also known as particle filter localization) is based on Bayes filter and importance sampling theories to estimate the robot position from sensor measurements and environment map. The most likely robot poses are represented by a set of particles that is iteratively updated as new sensor observations are obtained. In each iteration, those particles that produce a better matching of the measurement with the map are kept in the set. In general, occupancy grid maps are used in this method.

In a probabilistic way, MCL estimates the probability density function (PDF) of the posterior of the robot pose distribution $p(x_k|z_{1:k}, m)$, where the robot position is represented by x_k , the robot measurement is represented by z_k , the time stamp as k and the environment map as m .

The MCL algorithm starts by randomly distributing particles over the environment. The density of $p(x_k|z_{1:k}, m)$ is represented by the particle set $S_k = \{s_k^i\}_{i=1}^n$, where n is the number of particles and k represents the time stamp. Each particle s_k^i stores the position x_k^i (e.g. latitude, longitude, and orientation) and the importance weight w_k^i that represents the position certainty.

In each MCL iteration, S_k is updated through an auxiliary particle set $S'_k = \{s_k^i\}_{i=1}^n$.

Three steps comprise the particle set update:

1. Prediction step: The motion update step estimates the position x_k^i of particles s_k^i . The position is calculated by the probability $p(x_k|x_{k-1}, u_{k-1})$, where x and u corresponds to the robot position and the robot motion. Two motion update models are generally used: velocity model and odometry model. The first one is less precise by estimating the position from the velocity and the second one estimates the position from the raw pose measurements. The motion may be obtained by wheel encoder, IMU or GPS sensor. The estimated density is given as:

$$\hat{p}(x_k|z_{1:k-1}, m) = \sum_{i=1}^n p(x_k|x_{k-1}^i, u_{k-1}). \quad (5.1)$$

2. Update step: The measurement update step incorporates the sensor data z_k and map m to calculate the posterior density $\hat{p}(x_k|z_k)$. This step calculates the likelihood of s_k^i generate the z_k given m :

$$\hat{p}(x_k|z_{1:k}, m) \propto p(z_k|x_k, m) \hat{p}(x_k|z_{1:k-1}, m). \quad (5.2)$$

Importance sampling approximates an unknown distribution $f(x)$ from an approximated distribution $g(x)$ by weighting each sample with the term $w = \frac{f(x)}{g(x)}$. Using importance sampling to estimate the density we have:

$$p(x_k|z_{1:k}, m) = \frac{p(z_k|x_k, m) \hat{p}(x_k|z_{1:k-1}, m)}{\hat{p}(x_k|z_{1:k-1}, m)} = p(z_k|x_k, m). \quad (5.3)$$

The likelihood or weight of particle s_k^i is then given as:

$$w_k^i = p(z_k|x_k^i, m), \quad (5.4)$$

3. Resampling step: From S_k' is chosen randomly (with reposition) n particles. The probability of selection of a particle is proportional to its weight. Consequently, particles with higher weights have higher chances to be selected.

Figure 14 illustrates an example of MCL performing global localization in an environment with several rooms and ambiguous geometry features. In global localization, the particles are initially uniformly distributed over the map (figure 14a). After few iterations, the particles get concentrated in two clusters (figure 14b) (hypothesis) due to ambiguous measurements and finally converges into a single one (figure 14c) after getting few more observations.

In order to adapt the MCL to the GPOM, it is necessary to define how the measurement update will be performed, specifically, we have to model the likelihood model of equation 5.4 which depends on the type of the measurement and the map structure. The next subsections discuss the likelihood model calculation for OGM and GPOM representations.

Figure 14 – Example of MCL executing inside an office environment containing several rooms. (a) At the beginning of MCL, particles are uniformly distributed over the environment. (b) In ambiguous situation (measurements have been similar at more than one location), the particle set keep multiple hypothesis of the position. (c) The particle set concentrates at a single location after discarding the incorrect hypotheses by gathering more sensor measurements in the subsequent MCL iterations. Source: (DELLAERT et al., 1999).

5.2 Likelihood for Occupancy Grid Map

The update step of MCL determines the particles' likelihood to represent the actual pose and has a direct effect on which particles will remain in the set. The likelihood express the particle weight w_k^i . Essentially, the w_k^i gives the degree of correspondence between the measurement and the map, given the particle position through the probability $p(z_k|x_k^i, m)$. In this sense, the sensor (or observation) measurement model is used to calculate w_k^i .

Two sensor measurement models are commonly employed for laser-based sensors: beam range sensor and likelihood field. The first one ray-casts the laser beam in the map until the occurrence of an obstacle. The second one calculates the closest obstacle from the laser end point. Likelihood field model is preferred, because of the higher precision and the fact to be computationally more attractive (compared to ray-casting) (PFAFF; PLAGEMANN; BURGARD, 2007).

In likelihood field model, the probability $p(d^2, \sigma^2)$ is calculated for each laser end point through a normal distribution:

$$p(d^2, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{d^2}{2\sigma^2} \right), \quad (5.5)$$

where d and σ are the distance to the closest obstacle and the measurement uncertainty, respectively.

This generates a "Gaussian field" in the space. Specifically, regions closer to an obstacle are associated with larger values and regions farther from an obstacle with lower values. Following this idea, the likelihood of a single measurement is given by the following multiplicand:

$$p(z_k|x_k, m) = \prod_i p(d_i^2, \sigma^2), \quad (5.6)$$

where i is the beam index.

Figure 15 – Example of the range measurement likelihood field representation of an arbitrary environment. Darker regions corresponds to smaller likelihood values.¹

To determine which cells of the map are obstacles during the likelihood field model computation, occupancy probability values of OGM should be thresholded beforehand to classify each cell as free or occupied. Therefore, the likelihood is calculated from the discretized values of the OGM cells (i.e. binary occupancy grid map).

In practice, the likelihood of all cells of the OGM can be computed in advance and the result stored in a matrix. By doing this, the likelihood calculation during the MCL is simplified to a sequence of matrix access and multiplication.

5.3 Likelihood for Gaussian Process Occupancy Map

In our measurement likelihood function, occupancy and geometric information are used. First, to get the occupancy information, laser end point poses are obtained for each particle s_k^i , given its position $(x_k^i, y_k^i, \theta_k^i)$ and the current measurement $z_k = \{(r_k^j, a_k^j), j = 1, 2, \dots, n\}$:

$$p_k^i = \{p_k^{i,j}; j = 1, 2, \dots, n\}, \quad (5.7)$$

$$p_k^{i,j} = x_k^i + r_k^j \cos(a_k^j + \theta_k^i), y_k^i + r_k^j \sin(a_k^j + \theta_k^i), \quad (5.8)$$

where r_k^j and a_k^j denote the measurement z_k range distance and beam angle of beam index j , respectively, and n is the number of beams. We highlight that i corresponds to the particle index and $p_k^{i,j}$ is the end point position of beam j .

From the list of laser end point poses p_k^i , occupancy mean and variance are obtained

¹Image source: <http://ais.informatik.uni-freiburg.de/teaching/ss10/robotics/slides/07-sensor-models.pdf>

from each laser beam end point through GP prediction:

$$\text{occupancy}_k^i = \begin{bmatrix} \mu(p_k^{i,1}), \sigma(p_k^{i,1}) \\ \mu(p_k^{i,2}), \sigma(p_k^{i,2}) \\ \vdots \\ \mu(p_k^{i,n}), \sigma(p_k^{i,n}) \end{bmatrix} \quad (5.9)$$

$$m_k^i = \begin{bmatrix} \mu(p_k^{i,1}) \\ \mu(p_k^{i,2}) \\ \vdots \\ \mu(p_k^{i,n}) \end{bmatrix}, \quad v_k^i = \begin{bmatrix} \sigma(p_k^{i,1}) \\ \sigma(p_k^{i,2}) \\ \vdots \\ \sigma(p_k^{i,n}) \end{bmatrix} \quad (5.10)$$

where m_k^i and v_k^i corresponds to the mean and variance values obtained from p_k^i .

Using solely occupancy information can easily lead to ambiguous positions because it results in particles headed to any obstacle. Therefore, the geometric information of the laser measurements has been included into the likelihood model. We used the distance d_k^i between the particle to the first point along the laser beam direction which has occupancy mean higher than 0.99. Essentially, a laser measurement is simulated from the particle pose to determine how far the particle is to the obstacle. For this, ray-casting is employed in each laser beam.

$$d_k^i = \begin{bmatrix} d_k^{i,1} \\ d_k^{i,2} \\ \vdots \\ d_k^{i,n} \end{bmatrix} \quad (5.11)$$

where, $d_k^{i,j}$ is the distance from s_k^i pose to the closest obstacle along the direction given by beam j .

From this, it is possible to obtain the error ϵ_k^i between the measured range and the distance d_k^i in terms of squared difference:

$$\epsilon_k^i = \begin{bmatrix} (r_k^1 - d_k^{i,1})^2 \\ (r_k^2 - d_k^{i,2})^2 \\ \vdots \\ (r_k^m - d_k^{i,n})^2 \end{bmatrix} = \begin{bmatrix} \epsilon_k^{i,1} \\ \epsilon_k^{i,2} \\ \vdots \\ \epsilon_k^{i,n} \end{bmatrix} \quad (5.12)$$

Two arrays are formed by putting together the occupancy and the distance information associated with particle s_k^i :

$$\mu_k^i = \begin{bmatrix} m_k^i \\ d_k^i \end{bmatrix}, \quad \sigma_k^i = \begin{bmatrix} v_k^i \\ \epsilon_k^i \end{bmatrix}, \quad (5.13)$$

where μ_k^i and σ_k^i consolidates the mean and variance values extracted from the measurement z_k .

Finally, to calculate the particle weight w_k^i , it is calculated the likelihood to obtain the values of μ_k^i from the distribution given by the measurement z_k . This is done by connecting the measurement and the particle information in the multivariate normal probability density function:

$$w_k^i = p(z_k | x_k^i, m) \quad (5.14)$$

$$= p(z_k' | x_k^i, \mu_k^i, \Sigma_k^i) = \frac{1}{2\sqrt{|\Sigma_k^i|}} \exp \left\{ -\frac{1}{2} (z_k' - \mu_k^i)^T \Sigma_k^{-1} (z_k' - \mu_k^i) \right\}, \quad (5.15)$$

where, Σ_k^i is the covariance matrix with σ_k^i as the diagonal values and z_k' contains measurement values extracted from z_k . The matrix representation of Σ_k^i and z_k' is given as follows:

$$z_k' = \begin{bmatrix} U_n \\ r_k^i \end{bmatrix}, \quad \Sigma_k^i = \text{diag} \begin{bmatrix} v_k^i \\ \epsilon_k^i \end{bmatrix}, \quad (5.16)$$

with $U_n = [1, 1, \dots, 1]^T_n$ as a unitary array with n rows and $r_k^i = \{r_k^j\}_{j=1}^n$ as the range measurements. The unitary array stores the occupancy values of the end points of z_k (i.e. all end points fall in an occupied area).

5.4 Multimap Likelihood

In some situations, a robot may have two or more maps of the same environment and each one containing different information. For instance, an occupancy map built with measurement of the wall and another built with ceiling measurements. Depending on the situation, the use of several maps may result in a more robust localization, since this setting can make the localization suitable to noisier sensor measurements (DEUSCH et al., 2014; SUHR et al., 2016; ROHDE et al., 2016).

Technically, the particle weight calculation follows the same procedure as described before, but with the inclusion of a product operand:

$$w_k^i = \prod_j \eta_j p(z_k | x_k^i, m_j), \quad (5.17)$$

where m_j is the j -th map and η_j the normalization constant that defines the degree of importance of the likelihood given by map m_j . The product term implies in the independence between the maps.

The multi-map approach also enables mixing different map representations into a single likelihood model, such as GPOM and OGM. In order to facilitate this process, it is recommended to use the same coordinate reference in all maps for a straightforward access to the occupancy values.

5.5 Position Prior

The prior of the robot position $p(x_0)$ in MCL can be defined in two ways, named global localization and tracking. As the chosen prior model has a direct effect on the final result of the MCL, it must be defined carefully according to the localization problem.

Global localization defines the robot initial position distribution over all the environment. This implies in starting MCL by uniformly distributing the particles over the free areas of the map. The global localization approach is interesting when the initial position of the robot is unknown before MCL. In this sense, the number of particles should be sufficient to fill properly the environment. Typically, global localization is adopted in indoor localization problems.

On the contrary, in tracking, the starting position of the robot is given beforehand. Moreover, providing an approximate position is enough for tracking, as MCL generates a normal distribution with a certain variance around this place. In this approach, the initial position may be retrieved by an absolute position device, such as GNSS. Commonly, tracking is employed in outdoor localization problems.

5.6 MCL Variations

There are two popular extensions of the MCL method: augmented MCL and Kullback-Leibler divergence (KLD) MCL.

The augmented version is intended to enable the MCL to recover when the particles converge to a wrong location or the robot is teleported to another position. After the particle get concentrated in a single cluster, the standard MCL can't split it to make a subgroup of particles go towards the correct position. The augmented MCL solves this problem by randomly adding particles based on the average weight of the particles.

The KLD MCL adjusts the number of particles according to the particle set distribution. This makes the MCL computationally more attractive by making possible to reduce the particle set size as necessary. Essentially, KLD MCL checks during the particle resampling step if there are enough particles around a certain area. As consequence, particles are not added when there are sufficient concentration of particles.

Despite the popularity of these MCL variations, the current work focused on the standard MCL, in order to check the direct influence of the sparse maps on the localization.

5.7 Localization in Urban Environments

In our localization solution, first, the metric map containing urban structures is built and then, this map is used as a reference to correct the vehicle localization through Monte Carlo localization (MCL). Therefore, this localization system is comprised of two main parts:

environment mapping and vehicle localization. (The concepts related to robot mapping and localization have been discussed on [chapter 4](#) and [chapter 5](#)). These two are performed in totally separated steps, thus through offline processing. Next subsections provide details of the urban environment mapping and vehicle localization methods.

5.7.1 Urban Environment Mapping

Despite, theoretically not having the vehicle localization already working in this step, the mapping process depends on an accurate position source to make possible build a map from the sequence of curb and road marking clouds. A way to solve this problem is by applying Simultaneous Localization and Mapping (SLAM), but here we relied on a Global Navigation Satellite System (GNSS) device enabled with Real Time Kinematic (RTK) to facilitate this process. This technology uses a fixed antenna to reduce standard GNSS measurement errors to millimeter level. Hence, detection results are converted to Universal Transverse Mercator (UTM) coordinate. As bidimensional maps are used, the height field of UTM data is automatically discarded.

Before building the environment maps, it is necessary to define which information to store in, as several are available in urban environments. In this way, two road structures were stored in the map: curbs and road markings. These structures were chosen, mainly because of constant presence on the roads and hardly change along the time. (More information regards curb and road marking detection are present on [chapter 3](#).) Each feature is represented by a distinct map, thus, resulting in a curb map and a road marking map. As a consequence, curb and road marking observations are processed in a different way during the mapping process. In the case of curb data, it was generated OGM and GPOM representations which are used for MCL method based on OGM and MCL method based on GPOM, respectively. Road markings were represented just by OGM representation, due to the use as supplementary support during localization.

To build the curb maps, the resulting point clouds from the curb detection are converted into laser scan data. The conversion is necessary to obtain the free and occupied space along the roadsides. For this, equally spaced radial segments that cover 360° are projected from the center of the curb point cloud in the same process as ray-tracing. Segments that hit a curb point form the laser scan data. The set of laser scan data generated from curb points is georeferenced and used as the curb mapping dataset. This dataset is later reduced to remove redundant measurements through information theoretic data compression (details in [section 4.5.2](#)).

The curb mapping dataset is then fed to occupancy grid mapping and GP mapping methods. For occupancy grid mapping, the environment is reconstructed on a fine resolution grid (cell size lower than 0.30 m). Regards to GP mapping, the mapping dataset (also referred as training dataset) is clustered to form a set of GP experts as mentioned in [section 4.5.3](#). Essentially, each GP expert is associated with a subset of the training data. All GP experts are set

Figure 16 – Environment mapping diagram. The mapping process takes the GNSS coordinates and laser sensor as inputs and generates GPOM from curb data and OGM from road marking data. Two mapping approaches were employed to represent curb features.

with Matérn 3/ 2 covariance function and the same hyperparameters.

Due to the high processing cost of GP inference ($\mathcal{O}(n^3)$), the occupancy of the environment was also pre-computed to make the localization computationally tractable. For this, uniformly spaced points that cover all the environment are used as the GP test dataset. The occupancy mean and variance of each evaluated test point were then stored in a grid. Although the concept of GPOM is the continuous representation of the space, here we used a fine-grained grid (0.10 m resolution grid) as GPOM.

Regards to the road marking data, it was represented through an OGM. The mapping process of this feature consists of projecting the georeferenced output point cloud of the road marking detector into a high-resolution grid (0.10 m size cell). This produces a binary grid with value 1.0 in cells that represent road marking data.

In [figure 16](#) is illustrated the diagram that describes the mapping approaches. The main distinction between the approaches is the use of either OGM and GPOM to represent curb features.

5.7.2 Vehicle Localization

For vehicle localization, the detected curb and road marking data are matched to the urban map by the MCL in order to estimate the corrected vehicle position. Because MCL depends on a map, the localization is possible only inside the map boundaries.

In MCL, curb features are designated to correcting the lateral localization error as they are situated at roadsides, though are generally longitudinally redundant (e.g. straight roads). On the contrary, road marking features are intended to correct the longitudinal localization error due to the presence of structures that varies along the road extent (e.g. dashed markings, pedestrian crossing markings). As curb data is represented either as OGM or GPOM, an MCL based on

OGM (referred as MCL-OGM) and an MCL based on GPOM (referred as MCL-GPOM) were employed.

Here, we use the tracking approach for localization. For this reason, the MCL is started by setting the position prior $p(x_0)$ with a low precision GNSS measurement. Accordingly, the initial particle set configuration is defined by a normal distribution centered at the position prior. After MCL initialization, the GNSS measurement is no longer required.

In the prediction step of MCL, the motion model is employed to estimate the current position of the particles by taking into account only the vehicle motion. Specifically, odometry model approach is used as the motion model because of the higher accuracy compared to velocity model. Basically, the odometry model uses odometry data u_{k-1} (vehicle translation and rotation) gathered from left and right rear wheel encoders to predict the position of the particles by the probability $p(x_k|x_{k-1}, u_{k-1})$.

Then, the Velodyne sensor measurement z_{k-1} is decomposed into curb and road marking observations which are incorporated into position distribution for particle update. Therefore, the position distribution posterior is obtained by calculating the measurement likelihood. For curb data, there are two approaches for likelihood computation: a likelihood function that takes into account OGM (equation 5.6) and a likelihood function that takes into account GPOM (equation 5.15). Measurement likelihood based on road marking observation is computed exclusively through OGM formulation. The particle weight is calculated through multimap likelihood equation (equation 5.17) which put together curb and road marking likelihoods.

After computing the particle weights, the particle set is resampled as in the standard MCL algorithm. In order to calculate the vehicle position from the position distribution, the weighted average of the particles position is calculated:

$$x_k^* = w_k^i \sum_i x_k^i, \quad (5.18)$$

Figure 17 – Vehicle localization diagram. The MCL method receives laser data (curb and road marking point clouds), map and wheel encoder data to generate the position distribution posterior. The curb map leads to two localization approaches.

where x_k^* is the estimated vehicle position and x_k^i is the pose of particle s_k^i .

Figure 17 illustrates the diagram that summarizes the vehicle localization method developed in this work. MCL has a central role in integrating map and sensor data to estimate the position distribution posterior.

5.8 Final Remarks

MCL is a localization method widely employed in the robotics field. The main advantages are the possibility to represent the position by a multimodal distribution and the relative easiness to implement the algorithm. This makes suitable to adapt the original method for different goals. In this sense, an MCL method that makes use of GPOM has been presented. The following chapter presents the overall vehicle localization system that has the MCL with GPOM as a central role.

EXPERIMENTS AND RESULTS

This chapter describes the experiments performed to validate the localization method based on GPOM and the road feature extraction methods. Experiments were separated into indoor and outdoor environments (i.e. urban areas). The indoor experiments were intended to check the localization behavior in smaller areas before performing tests in larger areas. In outdoor experiments, we conducted tests for urban feature detection methods (curb and road marking detection), environment mapping of larger areas through OGM and GPOM structures, and evaluation of the OGM and GPOM based localization methods in urban roads.

All experiments were performed through offline processing. Therefore, datasets containing sensor data gathered in a previous step were post-processed for feature detection, mapping and localization experiments.

Next sections present details of the experiments and the obtained results. The robot and the autonomous vehicle setting is additionally described.

6.1 Experiments in Indoor Environments

Experiments in indoor places were aimed to validate the occupancy grid mapping and GP mapping methods to represent smaller areas, without the need to extract features from sensor measurements. Sparse laser readings were employed to produce unobserved areas. Later, the developed MCL methods were tested. In MCL based on OGM was verified the use of sparse maps for localization. Regards to MCL combined with GPOM, it was verified the functionality of the proposed measurement likelihood function.

Indoor experiments were performed through Freigurb (Dirk Haehnel)¹ and Rawseeds² publicly available robot datasets. These datasets are essentially formed by single beam laser

¹Source: <http://www2.informatik.uni-freiburg.de/stachnis/datasets/>

²Source: <http://www.rawseeds.org/home/category/benchmarking-toolkit/datasets/>

Scenario (dataset)	Traversed Length	Map Size	OGM Resolution
Seattle (Freiburg)	190 m	54.80 m × 19.75 m	0.10 m
Belgioioso (Freiburg)	213 m	93.25 m × 24.94 m	0.10 m
Bicocca (Rawseeds)	956 m	136.21 m × 161.95 m	0.10 m

Table 2 – Details of the datasets used in the indoor mapping and localization experiments.

sensor readings (laser scan data), encoder odometry data and position ground truth. The data is collected by a Pioneer robot with a Sick laser sensor mounted on the base. We used Seattle and Belgioioso datasets of Freiburg database, and Bicocca dataset of Rawseeds database. These datasets were gathered in indoor areas formed by static obstacles, such as walls, doors, and furniture. Particularly, the Bicocca scenario contains additional elements such as chairs, tables, bookcases, stairs and open areas. Further information of the datasets are presented in [table 2](#).

To test the map construction from sparse laser measurements, the dataset was intentionally reduced regarding the number of sensor readings and laser resolution. All measurements were limited to use up to 22 laser beams (approximately 1/8 of a full measurement). Moreover, for Seattle and Belgioioso, frames with even index were used and resulting in a mapping dataset with 241 and 132 laser measurements, respectively. As the Bicocca scenario is considerably larger than other datasets, we also applied information theoretic data compression which additionally removed 39.97% of the laser beams, resulting in a mapping dataset with 531 laser measurements. The same data was used for occupancy grid mapping and GP mapping experiments. OGMs were built with 0.10 m resolution grid.

As Seattle and Belgioioso are relatively smaller scenarios, a single GP expert was used for GP mapping. Resulting maps for Seattle and Belgioioso are illustrated in [figure 18](#). For Bicocca scenario, 15 GP experts were employed (see [figure 18b](#)). GP experts were obtained by clustering the dataset through k-means method (details on [section 4.5.3](#)). All GP models were configured with Matérn 3/2 covariance kernel. The obtained maps for Bicocca are shown in [figure 19](#). By visually analyzing the generated maps, GP mapping resulted in representations closer to the ground truth than OGM. Besides that, the resulting GPOMs could estimate the occupancy of unseen areas.

For localization experiments in the Seattle and Belgioioso scenarios, the original dataset was configured to return 180 laser beams per measurement (1 degree resolution). In the case of Bicocca scenario, we employed a dataset in which the robot travels a different path from the mapping experiment. The localization dataset of Bicocca was set to return 90 laser beams per measurement (2 degree resolution).

Robot localization was performed through MCL configured with 1000 particles and initially, the particle set was uniformly distributed over the map (i.e. global localization). To simplify, the MCL combined with OGM, and MCL combined with GPOM is referred as MCL-OGM and MCL-GPOM, respectively. For MCL-OGM, we employed the laser likelihood field

Figure 18 – Maps built from sparse laser data of Seattle and Biccocca datasets. (a–b) Ground truth, (c–d) OGM and (e–f) GPOM. Blue color is associated to free areas and red color to occupied areas. Axes are in meters.

Figure 19 – Maps built from sparse laser data of Biccocca dataset. (a) Reference map, (b) laser data reduction trough information theoretic method (red lines denotes the removed beams) and data clustering trough k-means ($k = 15$), (c) OGM and (d) GPOM modeled by 15 GP experts. Blue color is associated to free areas and red color to occupied areas. Axes are in meters.

model (equation 5.6) generated from a 0.10 m resolution OGM. Regarding the MCL-GPOM, we employed the likelihood of equation 5.15 which takes into account the whole laser beam information. Localization results are expressed by the average of the absolute difference between

Map	x (m)	y (m)	θ (rad)	xy (m)
OGM – Seattle	0.0772 ± 0.0376	0.0535 ± 0.0088	0.0265 ± 0.0007	0.1105 ± 0.0430
GPOM – Seattle	0.0348 ± 0.0032	0.0272 ± 0.0018	0.0214 ± 0.0004	0.0517 ± 0.0042
OGM – Belgioioso	0.0472 ± 0.0032	0.0366 ± 0.0019	0.0242 ± 0.0007	0.0670 ± 0.0041
GPOM – Belgioioso	0.0235 ± 0.0010	0.0185 ± 0.0007	0.0178 ± 0.0004	0.0342 ± 0.0014
OGM – Bicocca	0.2549 ± 0.9405	0.3235 ± 5.7596	0.0383 ± 0.0059	0.4705 ± 6.6483
GPOM – Bicocca	0.0594 ± 0.0201	0.0587 ± 0.0246	0.0230 ± 0.0012	0.0978 ± 0.0420

Table 3 – Numerical results for global localization in indoor environments. Absolute average longitudinal (x), lateral (y), Euclidean (xy) and angular (θ) errors of Seattle, Belgioioso and Bicocca scenarios.

Figure 20 – Global localization results of the indoor datasets along the test trajectory. The estimated position is denoted by blue lines and the ground truth pose is denoted by red lines. Euclidean error and particle variance is also plotted along the robot path which are represented by blue circles and green ellipses, respectively. Axes are in meters.

the estimated pose and the ground truth pose (i.e. localization error) obtained along the trajectory. Specifically, the longitudinal (x), lateral (y), orientation (θ) and euclidean (xy) errors were calculated.

Estimated localization errors and corresponding variance values for MCL-OGM and MCL-GPOM are shown in [table 3](#). Values obtained before MCL convergence were discarded (first four MCL iterations). In all experiments, MCL-GPOM resulted in lower errors for all categories, compared to MCL-OGM approach. Precisely, considering the Euclidean error, MCL-GPOM obtained 53.21%, 48.96% and 79.21% lower errors for Seattle, Belgioioso and Bicocca scenarios, respectively. The reduction was higher for Bicocca, despite this scenario is larger and contains more obstacles than Seattle and Belgioioso. This shows that MCL-GPOM is less susceptible to sensor noises and map uncertainties than MCL-OGM.

The obtained localization errors are also presented along with the localization path in [figure 20](#). In this, the estimated localization poses are represented by a blue line and the position ground truth by a red line. To get a better visualization of the difference between the estimated and real pose values, the Euclidean error is plotted by blue circles whose radius size is proportional to the error value. The particle set variance is also exhibited in the plots through green ellipses. The variance shows the particle concentration and expresses the position distribution uncertainty at a certain location.

Experiments in indoor areas validated the functionality of occupancy grid mapping, GP mapping and the proposed measurement likelihood model for MCL-GPOM. Obtained results were fundamental to enable experiments in outdoor areas using autonomous car prototype which are presented in the next sections.

6.2 Experiments in Urban Environments – Setup

The Carina II autonomous vehicle platform used in urban experiments consist on a Pali o Weekend with automatic transmission presented in [figure 21](#). This platform was mechanically modified to be fully controllable by software (drive-by-wire). All sensors are connected to an industrial computer running Ubuntu operating system. Collected sensor data was stored in ROS bag format to later perform offline processing. Regards to the implementation language, road feature detection methods were coded in C++, mapping and localization methods were coded in Matlab language.

Sensors Employed in Urban Experiments

Carina II is equipped with several sensors. For instance, GNSS, IMU, radars, laser sensors, and cameras. For the experiments in this work, the following were employed:

- Velodyne HDL-32E: Multilayer rotary laser sensor composed by 32 beam emitters with

Figure 21 – Carina II autonomous vehicle prototype equipped with Velodyne HDL-32E sensor on the top (in gray color).

the capacity to return infrared remittance data. This sensor is used to collect tridimensional point cloud data of the vehicle surroundings;

- Septentrio AsteRx2eH PRO: RTK enabled GNSS and equipped with two rover antennas. In favorable weather conditions, it can provide the position with millimeter precision. The GNSS device is used to georeference the Velodyne point cloud data and obtain the position ground truth for localization experiments;
- Yumo 1024 P/R Rotary Encoder: Quadrature encoder with 1024 pulse per rotation resolution. One encoder is attached in each rear wheel. The wheel encoders are used to obtain vehicle odometry data during MCL prediction step;
- Garmin 18x: Low cost GPS receiver with position errors of up to 15 m. Employed to obtain the vehicle approximate initial position for MCL.

Software and Libraries Employed in Urban Experiments

Sensors are interfaced through ROS framework which makes possible manipulating them using high-level languages. Collected sensor data is recorded in a ROS file format that later is manipulated through offline processing. For data processing, a desktop computer configured with Intel(R) Core(TM) i7-4700MQ, 8 GB memory and Ubuntu operating system was used to run the experiments. The following libraries and frameworks were additionally used for data processing:

- Robot Operating System (ROS)³: Libraries for sensor manipulation and sensor dataset reading;

³Framework website: <https://www.ros.org>

- Point Cloud Library (PCL)⁴: Manipulation of point cloud data returned by Velodyne sensor;
- Matlab: Employed for GP mapping and occupancy grid mapping. Vehicle localization methods were also developed in Matlab language;
- GP Library⁵: Gaussian process library developed by University of Sydney, used for GP inference and GP mapping.

6.3 Road Features Detection

Road Features Detection comprises of curb and road marking detection that is later employed to urban mapping and vehicle localization. The detection methods rely only on point cloud generated by Velodyne laser sensor. Methods were tested with measurements gathered in urban roads of São Carlos city during moderate to low traffics.

6.3.1 Curb Detection

Before applying curb detection, it is necessary to determine the optimal values of parameter set $\theta_{\text{curb}} = \{\alpha, \beta, t_s, t_d\}$ through parameter tuning. Three urban scenarios have been

Figure 22 – Curb detection before (a–c) and after (d–f) parameter tuning of three scenarios: left turn (a, d), straight road (b, e) and right turn (c, f). Red points denote the detected curb points and the blue lines denote the LTS regression curve. The number of detected curb points increases after the parameter tuning.

⁴Library website: <http://pointclouds.org/>

⁵Library source: <https://bitbucket.org/LionelOtt/gp>

considered for estimation of these parameters: a straight road, a left turn, and a right turn. These scenarios before parameter tuning are illustrated in [figure 22](#). For each scenario, curb points were manually marked in a previous step to get the ground truth. In this sense, the parameter optimization searches values for θ_{curb} that generates detection output as close as possible to ground truth (mathematical formulation defined in [equation 3.8](#)).

The parameter tuning execution was limited to 500 iterations and 20 executions. The best execution ($f_{\text{tun}} = 23033$) obtained the following values for θ_{curb} : $\alpha = 0.113$, $\beta = 1.375$, $t_s = 0.124$ m and $t_d = 0.596$ m. [Figure 22](#) shows results of the curb detection with the optimized parameters. The curb detection mean squared errors (MSEs) of each scenario were 2.507×10^{-4} (left turn), 4.53×10^{-3} (straight road) and 1.532×10^{-3} (right turn). The parameters do not need to be retrained unless the tested environment is considerably different from typical urban environments.

Curb detections in other urban scenarios using the obtained parameters are also shown in [figure 23](#). These scenarios illustrate the curb detection in streets with different vehicles occluding the roadsides. This is possible mainly thanks to false-positive removal by LTS regression (denoted by the blue lines of [figure 23](#)).

Figure 23 – Curb detection in different situations. (a) Cars on both street sides. (b) Two cars parked on the road side. (c) Two distant cars on the road side. Red points denote the detected curb points and the blue lines denote the LTS regression curve.

6.3.2 Road Marking Detection

First, as road marking detection relies on infrared remittance intensity data which is miscalibrated by default, the intensity values were calibrated through deterministic intensity calibration method (LEVINSON; THRUN, 2010). The calibration result is illustrated in figure 24. After the infrared data calibration, noises are reduced and the contrast between road markings and asphalt is enhanced.

For road marking detection, the threshold values for our modified version of Otsu method were configured with $t_n = 0.90$ (maximum separability value), $t_T = 197$ (maximum intensity for road markings) and $t_P = 0.80$ (maximum cumulative sum) for road marking detection.

These parameters were obtained empirically, because differently from curb detection, road marking detection parameter domain is restricted to $[0; 1]$ (for t_n and t_P) and $[0; 255]$ (for t_T). Additionally, the detection result is less sensitive to slight changes in the parameter values.

Figure 25 shows the road marking detection in different urban streets. Blue lines represent the road boundaries obtained by the curb detector. These lines delimit the search space of the road markings. Yellow points represent the road marking detected by the adapted Otsu method.

The intensity histogram aspect of the road represented in figure 25(b) is shown in figure 26. The Otsu method divided the bimodal histogram at $T = 160$, which resulted in a maximum variance of $\sigma_L^2 = 27.13$. At this point, the cumulative sum and mean show that the number of points starts to increase at value T (i.e., the starting intensity value of the asphalt surface). The values of cumulative sum and separability measure were, respectively, $P(T) = 0.0128$ and $\eta(T) = 0.8589$, which satisfied t_P and t_n .

Figure 24 – (a)–(b) Factory original intensity. (c)–(d) Calibrated intensity which leads to a better contrast between asphalt and road markings. The point cloud was artificially colored according to the remittance intensity value for a better visualization.

Figure 25 – Road marking detected by a modified version of Otsu method. (a)–(d) Single frame result. (e)–(g) Accumulated results from consecutive frames. Blue lines represent the road boundaries estimated by the curb detector and yellow points correspond to the detected road marking.

Figure 26 – Intensity histogram of the LIDAR points that intercept road marking. The dashed line represents the obtained threshold.

6.4 Urban Environment Mapping

Resulting curb and road marking point clouds obtained from the detection methods were georeferenced (through GNSS with RTK) before mapping. From this, curb data was employed

to build Gaussian process occupancy map (GPOM) and road marking data to build occupancy grid map (OGM).

Two datasets were employed for mapping experiments, referred here as Track 1 and Track 2. These datasets consist of sensor data collected from streets inside the Campus 2 of USP São Carlos. Though each dataset contains sensor log of three laps, just one was used for mapping. The path covered by each dataset is denoted by the blue line in [figure 27](#). Track 1 is a 770 m long path and Track 2 is a 1749 m long path. Measurements were gathered during low traffic and the vehicle ran at approximately 40 km/ h. Complementary details of the datasets are presented in [table 4](#).

Experiments and results of curb mapping and road marking mapping are presented in the following sections.

Dataset	Length	Map Size	Resolution (OGM)
Track 1	770 m	199.9 m × 306.9 m	0.10 ~ 0.30 m
Track 2	1749 m	573.0 m × 536.7 m	0.10 ~ 0.30 m

Table 4 – Details of the tracks used in the mapping and localization experiments.

(a) Track 1 – Aerial Map

(b) Track 2 – Aerial Map

Figure 27 – Environment maps used in the localization experiments. Track 1 is 770 m long and Track 2 is 1749 m long.

6.4.1 Curb mapping

Before mapping, curb point clouds are converted into laser scan data in order to extract free and occupied space information. This allows representing the navigable area along the

road extent in the map. For this, it is simulated in the detected curb points, a 360° laser scan configured to return at maximum 90 laser beams (45 directed to front and 45 directed to back, corresponding to 1/4 of conventional laser scan sensor resolution). The set of laser scan data generated from curb points was georeferenced to form the mapping (training) dataset. This dataset was then reduced through information theoretic data compression method to remove redundant measurements (section 4.5.2). The resulting training dataset for Track 1 and Track 2 is formed by 17252 (39.88% reduction) and 11830 (40.03% reduction) laser beams (i.e. lines and points), respectively. Training data measurements are illustrated in figure 28(a)–(b).

It is highlighted that the curb detection itself returns a sparse point cloud and it is

Figure 28 – Training data set used to build GPOM and OGM. (a)–(b) Clustered mapping (training) dataset distinguished by a color. (c)–(d) OGM configured with 0.10 cm resolution. Blue and red colors are related to free and occupied areas, respectively. Axes are in meters.

Figure 29 – Output values of the GP mapping. The GPOM corresponds to the squashed mean values. (a)–(b) Mean value. (c)–(d) Variance Value. (e)–(f) Squashed mean obtained from mean and variance values. Blue and red colors are associated to free and occupied areas, respectively. Axes are in meters.

additionally reduced to induce the production of unseen areas. As the set of measurements of a single loop is used, the resulting mapping dataset is considerably sparse if considered the track extent.

The GP classifier kernel was set with squared exponential covariance and the hyper-parameters were trained with gradient descent method. To reduce the computational cost, the training dataset was clustered through k-means to form 15 GP experts (details on [section 4.5.3](#)). All GP experts use the same hyper-parameters for occupancy inference. The portion of dataset assigned to each GP expert is represented by a distinct color in [figure 28\(a\)–\(b\)](#).

To construct the GPOM, test points must be provided to the GP model to obtain the GP

Figure 30 – Amplified regions of Track 1 maps. Blue and red colors are associated to free and occupied areas, respectively. Axes are in meters.

Figure 31 – Amplified regions of Track 2 maps. Blue and red colors are associated to free and occupied areas, respectively. Axes are in meters.

mean and variance of the environment space. For this, a test dataset containing evenly spaced (x, y) points with 0.10 m spacing was used. The occupancy of each point is inferred by the closest GP expert.

The inferred mean and variance values are then fed to the squashing function to classify the occupancy of the query locations (equation 4.18). The resulting GPOM with the GP mean and variance values are presented in figure 29.

In order to validate the resulting GPOM, OGMs configured with 0.10 m, 0.20 m and 0.30 m resolution were built using the same curb data. These resolutions depict fine to moderate grained grids that are commonly employed in localization problems (MONTEMERLO et al.,

2008; LEVINSON et al., 2011). As we do not have the ground truth map, the resulting maps were evaluated through visual analysis.

In figure 28 and figure 29, the road area (blue color) and the curb obstacle (red color) are better highlighted in GPOM than in OGM. Moreover, GPOM defines a region of transition between occupied and free areas. Figure 30 and figure 31 presents details of the GPOM and OGM built with different resolutions. Labels A and B shows portions of the maps built with relatively denser measurements. It is noticed the presence of gaps in the OGM independently of the resolution, while GPOM generates a uniform surface. This situation degrades in maps with labels C and D, where there are visible unobserved areas, which results in large gaps in OGM, whereas GPOM could reconstruct the profile of most of the road.

6.4.2 Road marking mapping

Differently from curbs, road markings were represented exclusively through OGM because of the irregularity (in contrast to curbs, road markings are not present in every Velodyne frame) and they are used to provide auxiliary support during localization. Specifically, road markings were designated to provide supplementary information for measurement likelihood model at places that curbs fail to provide accurate position estimation.

The road marking mapping process consists on projecting the georeferenced road marking point cloud in an OGM. As a result, rather than using occupancy grid mapping algorithm, cells associated to road markings are directly marked as occupied. In this sense, the final map is a binary grid where the cells inform the presence or absence of road marking.

The obtained road marking maps for Track 1 and Tack 2 are shown in figure 32. Maps

Figure 32 – Occupancy grid maps with 0.10 m resolution containing road marking data. Axes are in meters.

(a) A – OGM (0.10 cm) (b) B – OGM (0.10 cm) (c) C – OGM (0.10 cm) (d) D – OGM (0.10 cm)

Figure 33 – Zoomed regions of road marking map of Track 1.

(a) A – OGM (0.10 cm) (b) B – OGM (0.10 cm) (c) C – OGM (0.10 cm) (d) D – OGM (0.10 cm)

Figure 34 – Zoomed regions of road marking map of Track 2. Axes are in meters.

were configured with 0.10 m cell resolution. For a better visualization of the road markings, amplified areas of the map are provided in [figure 33](#) and [figure 34](#).

6.5 Vehicle Localization in Urban Environments

As stated in previous chapters, the main purpose of map aided localization is to obtain an accurate position estimation without the dependence of expensive and high precise GNSS devices. The modified MCL method depends on urban maps (curb and road marking maps) and laser measurements to match the detected road features to the map. Besides the multilayer laser sensor, the method relies on a low-cost GNSS (Garmin 18x) to obtain the position prior and rear wheel encoders (Yumo rotary encoder) to estimate the odometry motion during prediction step of MCL. These sensors present low accuracy compared to position estimation provided by GNSS enabled with RTK solution. To exemplify, a single loop on Track 1 gives an average Euclidean error of 3.2456 m and 17.0274 m for low-cost GNSS and wheel encoders, respectively. Despite lower errors of Garmin GNSS and apparently matching Septentrio GNSS data, this device can produce lateral and longitudinal offsets larger than 10 m. [Figure 35](#) illustrates the position estimation of these sensors along the Track 1 path and [table 5](#) lists detailed values of the sensor errors.

In all experiments, MCL was configured to use 1000 particles. Here MCL combined with OGM and MCL combined with GPOM are referred as MCL-OGM and MCL-GPOM, respectively. Essentially, MCL-OGM experiments were aimed to evaluate the use of curb and

Figure 35 – Comparison of position sensors output obtained from Track 1 path. Taking the RTK GNSS (red points) as a reference, wheel encoder (blue points) accumulates position error as the traveled distance increases and the low-cost GPS (green points) gets shifted in parts of the track. Axes are in meters.

road marking features for vehicle localization. The MCL-GPOM experiments were focused on evaluating the use of GPOM representation for vehicle localization. Further, we performed a comparison between MCL-OGM and MCL-GPOM in terms of localization errors. The same datasets were used for MCL-OGM and MCL-GPOM tests.

We conducted localization experiments in scenarios of Tracks 1 and Track 2 during low traffic. The vehicle velocity varied from 30 to 45 km/h. In each track, we performed one experiment using only curb measurement and a second experiment using both curb and road marking measurements. To incorporate road marking observations in the curb measurement model, multimap likelihood model has been employed (equation 5.17). We highlight that the datasets designated to localization experiment are different from the mapping datasets.

The localization results are given by longitudinal (x), lateral (y), orientation (θ) and euclidean (xy) position offsets. These errors (offsets) are given by the absolute difference between estimated and ground truth positions (gathered by a GNSS-RTK device). The following sections describe the localization results obtained in MCL-OGM and MCL-GPOM experiments.

Sensor	Average Eucl. Error	Max. Long. Error	Max. Lat. Error
Garmin 18x	3.2456 m	14.3230 m	11.2217 m
Yumo Encoder	13.1047 m	23.6760 m	13.3097 m

Table 5 – Position errors of the low-cost GNSS and rear wheel encoders after traveled a single loop of Track 1. Axes are in meters.

Figure 36 – Laser likelihood field with 0.10 m resolution build using OGM containing curb data and OGM containing road marking data. Axes are in meters.

6.5.1 MCL using OGM

The MCL-OGM uses laser likelihood field function to update the particle weights (equation 5.6). In order to prevent unnecessary computations, the measurement likelihood values were calculated beforehand for each OGM cell and stored in a lookup table. Obtained likelihood values for OGMs formed by curb and road marking data are represented in figure 36.

In order to take into account the effect of grid resolution in the localization result, we performed experiments using curb OGM built with 0.10 m, 0.20 m and 0.30 m cell resolutions. The road marking OGM was fixed to 0.10 m cell resolution in all experiments. For each scenario, first, we employed experiments using only curb features and then experiments adding road