UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Environment Modeling for Autonomous Vehicle Applications

Permalink

https://escholarship.org/uc/item/37m9d59t

Author

Li, Qinru

Publication Date

2020

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA SAN DIEGO

Environment Modeling for Autonomous Vehicle Applications

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Qinru Li

Committee in charge:

Professor Henrik I. Christensen, Chair
Professor Manmohan Krish Chandraker
Professor Todd L. Hylton

2020

Copyright
Qinru Li, 2020
All rights reserved.

The thesis of Qinru Li is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California San Diego

2020

DEDICATION

To Mr. Yuqing Li and Mrs. Qiaofei Li, for their love;

EPIGRAPH

With the eyes of our own,

the eyes of machines,

the eyes of logic,

and the eyes of imagination,

we observe this world,

and search for truth.

—Cho-yun Hsu

TABLE OF CONTENTS

Signature Page iii

Dedication iv

Epigraph v

Table of Contents vi

List of Figures viii

List of Tables ix

Acknowledgements x

Vita xi

Abstract of the Thesis xii

Chapter 1 Introduction 1
1.1 The Mapping Problem 1
1.2 Overview of the Thesis 3

Chapter 2 Parametric Mapping Techniques 4
2.1 The Map Representation 4
2.2 Simultaneous Localization and Mapping 7
2.2.1 Problem Definition 7
2.2.2 Occupancy Grid Map 9
2.2.3 Landmark-based Map 13
2.3 Conclusion 17

Chapter 3 Semantic Mapping 18
3.1 Background 19
3.2 Proposed Method 21
3.2.1 Image Semantic Segmentation 21
3.2.2 Point Cloud Semantic Association 22
3.2.3 Semantic Mapping 23
3.3 Experiments and Results 25
3.3.1 Image Semantic segmentation 26
3.3.2 Semantic Mapping 27
3.3.3 Comparison to Sparse LiDAR Scan 30
3.3.4 Comparison to Planar Assumption 30
3.4 Conclusion 32

3.5 Acknowledgments 32

Chapter 4 Conclusion and Future Directions 33

Bibliography 35

LIST OF FIGURES

Figure 2.1: A comparison between the occupancy grid maps generated from dead-reckoning and the particle filter in the same environment. 12
Figure 2.2: A comparison between the landmark-based maps generated from dead-reckoning and the Extended Kalman filter in the same environment. 16

Figure 3.1: Our pipeline for generating probabilistic semantic maps for road feature extraction and HD mapping applications. 21
Figure 3.2: The map generation result of our method. The top image is the local map and the bottom image is the same region in the global map. We highlighted the ego-vehicle as a blue box on the local map. 24
Figure 3.3: Data collection vehicle sensor configuration 26
Figure 3.4: A visualization of our generated map, the ground truth label, and the LiDAR point cloud map. 30
Figure 3.5: The BEV version of our generated map in the entire region of our testing dataset. Our semantic map is displayed on top of the dense point cloud map. One segment of the map has been magnified to show the details of the map. Best view in color. 31
Figure 3.6: A semantic map generated from real-time LiDAR scan in the local frame. The black region on the scene indicates the unknown area which is not detected by the LiDAR sensor. The ego-vehicle is highlighted as a blue box on the map. 31
Figure 3.7: A local semantic map generated by back-projecting 2D semantic image into the 3D space with planar assumption. The map presented in the vehicle frame. The distortion of lines indicates the failure case. The ego-vehicle is highlighted as a blue box on the map. 32

LIST OF TABLES

Table 3.1: Training Labels and Their Associated Colors for the Semantic Segmentation Network 22
Table 3.2: Quantitative evaluation on our labeled data for road, crosswalk and lane mark regions. 28

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to Professor Henrik I. Christensen, my advisor and the chair of this thesis committee, for his guidance and insights towards the work contains in this thesis. Through many discussions and experiments, his advice has proven to be invaluable. The work here would not be realized without his profound knowledge and vigorous research methodologies.

I would like to thank Professor Manmohan Krish Chandraker for his advice and supports. He opens the door of deep learning technology for me. Many of his words are analogous to the say that a man will recall, while he is sitting in the middle of the mountains, from the elders in the previous village. I would also like to acknowledge Professor Todd L. Hylton for his invaluable instructions and criticisms. Many of his suggestions have extended from academia to philosophy and have been forged into paddles for me to sail ahead.

I would also like to acknowledge all the members of the AVL team, particularly David Paz, Henry Zhang, Hao Xiang, and members from Cogrob lab: Anwesan Pal, Shengye Wang, Yiding Qiu for their fascinating support on my master study. Without the magic they do in hardware and software, the idea presented in this work can never come true.

Last but not least, I would like to acknowledge my father Yuqing Li and mother Qiaofei Li, for their unconditional love.

Chapter 3 contains material as it appears in the International Conference on Intelligent Robots and Systems 2020. Paz, David; Zhang, Hengyuan; Li, Qinru; Xiang, Hao; Christensen, Henrik I., The thesis author was one of the primary investigators and authors of this paper.

VITA

2018	Bachelor of Science in Electrical Engineering, University of Illinois Urbana-Champaign
2020	Master of Science in Computer Science, University of California San Diego

PUBLICATIONS

Paz, David; Zhang, Hengyuan; Li, Qinru; Xiang, Hao; Christensen, Henrik I.. "Probabilistic Semantic Mapping for Urban Autonomous Driving Applications," International Conference on Intelligent Robots and Systems, 2020.

ABSTRACT OF THE THESIS

Environment Modeling for Autonomous Vehicle Applications

by

Qinru Li

Master of Science in Computer Science

University of California San Diego, 2020

Professor Henrik I. Christensen, Chair

Recent advancements in statistical learning and computational abilities have enabled autonomous vehicle technology to develop at a much faster rate. While many self-driving architectures previously introduced are capable of operating under highly dynamic environments, many of these are constrained to smaller-scale deployments, heavy dependence on high-definition (HD) maps that require constant maintenance and tedious manual labeling. This thesis is trying to address the mapping problem in the context of autonomous vehicle applications. We study the existing map representations in the literature and analyze the performance of parametric mapping algorithms in both indoor and outdoor environments. We then propose to fuse image and pre-built point cloud map information to perform automatic and accurate labeling of static landmarks such

as roads, sidewalks, crosswalks, and lanes. The method performs semantic segmentation on 2D images, associates the semantic labels with point cloud maps to accurately localize them in the world, and leverages the confusion matrix formulation to construct a probabilistic semantic map in bird's eye view from semantic point clouds. Experiments from data collected in an urban environment show that this model is able to predict most road features and can be extended for automatically incorporating road features into HD maps with potential future work directions.

Chapter 1

Introduction

1.1 The Mapping Problem

How to map the environment is one of the most important problems in the pursuit of building truly autonomous mobile robots [Thr02]. It is a fundamental skill robots need to localize themselves, perform navigations, and interact with humans in the real-world environment. Over the past few decades, people have studied various techniques to tackle this problem. From algorithms [DRMS07, KA08, MAMT15, MAT17] that simultaneously localize the robot position and mapping the environment as they travel along the spaces, to algorithms that can recognize complex scene environment both geometrically and semantically [SK02, WS08, TINGC10, PNGC19], the robotics community has observed robust mapping solutions in static, small-size, and structured environments [Thr02], but how to address this question in dynamic, large-scale, and more uncertain environments is still an open question.

The complexity of this question not only comes from the environmental challenges, but also from the map's definition. A map can be viewed as the robot's knowledge of the world. The metric map is probably the first kind of mapping technique that jumps into our mind when we talk about the mapping problem. Its accurate description of the environment in the metric

space provides an intuitive integration with robotics applications and examples like grid-based map have enjoyed great success in both industry and academia [Elf89, MM94]. However, if we look at animals, including us human beings, we don't rely heavily on these metric maps in our daily life [GM98, WS02, FWDT05]. This inspires us to think of that the robot's knowledge of the environment can be abstracted to contain higher-level information. Such abstraction offers two advantages. The first one is the efficiency, as we don't have to memorize all details of the environments. The second one is the robustness. While the data acquired from the physical sensors is subjected to noises, a map's sensitivity to these noises can be reduced by the levels of abstraction and hierarchy imposed on sensory measurements [PS17, GFLM17].

The challenge of utilizing these non-metric map stems from the difficulty to describe them in an exact mathematical form. For example, how to explain the idea of "There is a bump in the middle of the road and two bike lanes on the side." with a set of vectors? Fortunately, with the recent development in deep learning technology, we now may have a chance to encode these complicated concepts into digits by using machine learning algorithms. In the deep neural networks, conceptual knowledge is embodied by a stack of neurons [Ros58, LBH15], which are essentially a set of scale values, and the networks update values of all neurons based on the loss signal from the outputs of the networks [RHW86]. By constantly updating the values of neurons from ground truth labels, the neural networks can "learn" abstracted concepts gradually. We have observed a lot of great achievements from the computer vision community, where networks are trained to understand objects and their relationship in the scenes [GDDM14, WGGH18, YLL$^+$18] or capture the semantic information from 2D images [LSD15, ZSQ$^+$17, CPSA17].

The success of deep learning technology certainly galvanizes people to study the mapping problem from a new perspective [KG15, LSN$^+$16], and a newly emerged robotic platform has pushed this desire even further: the autonomous vehicle. In this platform, the ability to efficiently model large regions, to handle the scene dynamics, and to generalize well in unseen environments, are no longer somethings nice-to-have for a mapping algorithm but the key requirements to

deliver a safe, commercializable, and durable robotic vehicle that can transform people from one place to another. The maps for a fully automated vehicle should be able to describe not only the surroundings in metric space, but also convey semantic traffic rules that vehicles should obey on the road. It is a challenging yet exciting opportunity to study the mapping problem in a such robotic system.

1.2 Overview of the Thesis

How should we address the mapping problem in the context of autonomous vehicle applications? This is the very question that we want to study in this thesis. Starting from Chapter 2, we will examine the existing mapping techniques in the literature and their probabilistic frameworks that can improve the map's robustness to uncertainty and measurement noise. Their performance will be evaluated in the simultaneous mapping and localization (SLAM) problem because it contains common scenarios of mapping in unknown environments. In Chapter 3, we will propose a new solution inspired by these studies to generate semantic maps for autonomous vehicles automatically. The proposed method utilizes deep learning technology and the fusion between LiDAR and camera sensors to create a large scale semantic map in an outdoor environment without human annotations. In Chapter 4, we will summarize the lessons that we learn throughout these experiments and have a discussion for future directions.

Chapter 2

Parametric Mapping Techniques

Before we build any form of maps for robotic applications, there is one question we should ask: how should we represent and update the maps in the mathematical form? The methodologies to answer this question are the major contents of this chapter. We will first talk about some generic categories of the map representations: the metric maps, non-metric maps, and their fusion (hybrid maps). A probabilistic framework known as Bayes' filter will be discussed to update the maps in the context of the SLAM problem. We will use two variants of the Bayes' filter to map the indoor and outdoor environments. The mapping techniques introduced in this chapter will serve as the cornerstones of the next chapter.

2.1 The Map Representation

Rooted in the rich study of geometry in mathematics, metric representation is one of the most popular map representations that can be easily integrated into many robotic tasks. The understanding of how to choose a proper metric model for the environments will impact many research areas, including localization and navigation. The occupancy grid map is one of the predominant methodologies in this metric family. Developed by Elfes and Moravec [ME85, Elf89] in the late 1980s, the occupancy grid maps discretize the environments into small,

uniformed regions called cells and keep track of the probability of each cell being occupied. For the numerical stability reasons, the probabilities are usually written in a log-odd form [Mor88]: given a cell c_i, its log-odd value $\lambda(c_i)$ is

$$\lambda(c_i) = log\frac{P(c_i = 1)}{P(c_i = 0)}$$

where $c_i = 1$ means the cell is occupied and $c_i = 0$ otherwise. With a little application of Bayes' rule, the updates of this log-odd value are incremental, which is mathematically convenient to process new data (We will show a more detailed example in the Section 2.2.2.). However, the corresponding cost is the map's disability to model the correlation between cells since each of them is updated independently. Another shortcoming of the occupancy grid maps relates to their resolutions. The higher the resolution of a grid map is, the smaller region each cell covers, and thus the more storage in memory it requires.

A direct solution to avoid excessive memory consumption would be the usage of a sparse metric map, which is known as landmark-based map, or feature-based map. Maps in this category use a set of sparse 3D landmarks to represent the environments [CCC$^+$16]. This approach is very popular in the SLAM community and computer vision community [SLL02, YML05, SLL05], and can achieve real-time performance with a single camera [DRMS07, MAMT15]. To avoid perceptual aliasing, a phenomenon that happens when the mapping algorithm cannot distinguish two different landmarks apart [CN08, SAH04], each landmark usually comes with a unique descriptor (e.g. a feature vector) to aid the mapping algorithm associate different sensory measurements. While a large amount of works have been conducted on the point-based landmark maps, researchers have extended their works to more sophisticated geometric and visual features of the scenes, such as line [LL07], edge [ED09], surface [TINGC10], and appearances [CN08, CN11].

Non-metric map like topological map is another alternative method to represent the

environment. Instead of collecting accurate metric information from the scene, it "leverages place recognition to build a graph in which the nodes represent places and edges represent the reachability among places" [CCC+16]. The similarity between our human's conceptual graph of the environment and the topological map makes it suitable for tasks that have humans in the loop [TC06, AC03]. However, the granularity of the places is often task-specific and is hard to generalize well in unknown environments [DTK05, NPB95]. The gap between the abstracted routes to places and the physical metric spaces [LSN+16] is also a challenge of building topological maps and a careful association such as [Kui00] between these two parts has to be defined for meaningful robot actions in the physical worlds. For the same purpose as landmark-based maps, each node (i.e. place) needs a descriptor to distinguish itself from others. A curious reader can find a comprehensive study in this area from Lowry's survey [LSN+16].

There certainly is a desire to combine the benefit from both metric map and non-metric map together, and semantic mapping is one of such examples. It fuses the geometric constraints from metric space with the semantic concepts of topological maps [KG15, WS08]. The observation is that the association of semantic concepts with different parts of the environments can improve robot's robustness, facilitate more complex tasks, and enable advanced human-robot interaction [CCC+16, Cha12]. There are, however, no strict notions on how the concepts should be designed. The definition and structure of concepts are often task-specific. For example, the map that Pronobis et.al. [PJ12] generate contains shape, size, appearance, and object's functions in the indoor environment, whereas Sengupta et. al. [SGST13a] utilize a completely different semantic hierarchy for their outdoor map.

Researches have studied many other combinations of different map representations which improve the resulting environment models [Thr98, FS00, PJ12]. It is a trend that as the tasks become more and more complicated, a single metric map or non-metric map is insufficient. A map representation that contains information from both sides of the world is most likely to help the robot achieve full autonomy in the future.

2.2 Simultaneous Localization and Mapping

The relationship between localization and mapping is analogous to the one between chicken and egg. In order to model the environment, we need to know the robot's pose. In order to determine the robot's pose, we need to understand its surrounding environments. In our previous discussion of map representation, we have implicitly assumed that the pose of the robot is known. In fact, it is rarely the case. Usually both the robot's localization and environment are unknown. In this situation, it is particularly beneficial to estimate them in conjunction because by focusing just on one the other will introduce systematic noise [Thr02]. Such kind of problem is known as the simultaneous localization and mapping (SLAM) problem. Conventionally, an algorithm that solves the SLAM problem is called a SLAM algorithm.

In this section, we will use SLAM algorithm to show how to generate and update the metric maps in a probabilistic framework. The probability is used to capture the uncertainty of our estimation of both the environment and pose. A formal definition of the SLAM problem will first be established. Then, under this framework, we will show an example of occupancy grid map generated from an indoor dataset and a landmark-based map generated from an outdoor dataset.

2.2.1 Problem Definition

Given a sequence of control signals $u_{0:T}$ and a sequence of observations of the environment $z_{0:T}$ from time 0 to time T, the SLAM algorithm aims to find the best estimation of the robot's trajectory $x_{0:T}$ and a map of the environment m, as presented below:

$$\max_{x_{0:T},m} P(x_{0:T}, m | u_{0:T}, z_{0:T}) \tag{2.1}$$

Assuming that the measurement of the map at each time step is conditionally independent given the robot trajectory, we can simplify this function as

$$P(x_{0:T}, m|u_{0:T}, z_{0:T}) = P(x_{0:T}|u_{0:T}, z_{0:T}) \prod_{t=0}^{T} P(m_t|u_{0:T}, z_{0:T}, x_{0:T}) \tag{2.2}$$

Our objective function now has been decomposed to $T+1$ subproblems: one for estimating the posterior of robot's trajectory $P(x_{0:T}|u_{0:T}, z_{0:T})$, and T for estimating the map conditioned on the robot's trajectory $P(m_t|u_{0:T}, z_{0:T}, x_{0:T})$. For the map probability conditioned on the robot's trajectory at each time step, we can simplify it as

$$P(m_t|u_{0:T}, z_{0:T}, x_{0:T}) = P(m_t|z_{0:T}, x_{0:T}) \tag{2.3}$$

based on Markov assumption.

The posterior of robot's trajectory at any time $t+1$ can by estimated by Bayes filter which has prediction and update steps. In the prediction step, we estimate robot pose $x_{0:t+1}$ given observation $z_{0:t}$ and control input $u_{0:t}$. In the update step, we correct our previous prediction based on new observation $z_{0:t+1}$. The probability that we need to estimate at each step is summarized below.

$$\text{Prediction:} P(x_{0:t+1}|z_{0:t}, u_{0:t})$$

$$\text{Update:} P(x_{0:t+1}|z_{0:t+1}, u_{0:t})$$

As the focus of this chapter is on the mapping techniques, we loose our requirement of this framework and only requires to find the best robot pose and map of environments at time t. In other words, our objective function becomes

$$\max_{x_t, m} P(x_t, m|u_{0:T}, z_{0:T}) \tag{2.4}$$

This is essentially an online version of the SLAM [TBF05]. In this case Bayes filter's update and prediction rules become

$$\text{Prediction:} P(x_{t+1}|z_{0:t}, u_{0:t})$$

$$\text{Update:} P(x_{t+1}|z_{0:t+1}, u_{0:t})$$

2.2.2 Occupancy Grid Map

We use a 2D occupancy grid map to model an indoor environment collected by humanoid robot THOR [YMV+14]. The robot has equipped with an IMU, a LiDAR sensor, and several rotational encoders recording the angles of each joint. The IMU sensor returns the x, y, z coordinates as well as the roll, pitch, yaw angles of the robot with respect to its initial position. LiDAR sensor will generate 1081 points per frame ranging from -135 degree to 135 degree (the center of LiDAR sensor corresponds to 0 degree.) with an angular resolution of 0.25 degree. The reading of a LiDAR point is a single scale which reflects the distance between obstacles and the sensor. All the data from different sensors is synchronized with respect to LiDAR's time stamp and transformed into the same coordinate frame: the center of the mass of the robot.

The occupancy grid map keeps track of the probability $P(m_t \mid z_{0:t}, x_{0:t})$ at each time t. It digitalizes an area of 40×40 square meter into a 801×801 grid, presented in the bird's eye view. The resolution of a cell in the map is 0.05×0.05 meters squared. As we mentioned in Section 2.1, cells here are viewed as independent Bernoulli random variables. Now with slightly different annotations which will simplify our expressions later on, we denote the probability of a cell c_i being occupied as $\gamma_{i,t} = P(c_i = 1 \mid z_{0:t}, x_{0:t})$ and the probability of being free as $1 - \gamma_{i,t}$. The odds ratio is

$$o(c_i \mid z_{0:t}, x_{0:t}) = \frac{P(c_i = 1 \mid z_{0:t}, x_{0:t})}{P(c_i = -1 \mid z_{0:t}, x_{0:t})} = \frac{\gamma_{i,t}}{1 - \gamma_{i,t}} \tag{2.5}$$

Taking the log of this ratio, we can derive the following update rule for the cell's proba-

bilistic distribution:

$$\begin{aligned}\lambda(c_i \mid z_{0:t}, x_{0:t}) &= \log\left(o(c_i \mid z_{0:t}, x_{0:t})\right) \\ &= \log\left(\frac{P(z_t \mid c_i = 1, x_t)}{P(z_t \mid c_i = -1, x_t)} \frac{\gamma_{i,t-1}}{1-\gamma_{i,t-1}}\right) \\ &= \log\left(g_h(z_t \mid c_i, x_t)\, o(c_i \mid z_{0:t-1}, x_{0:t-1})\right)\end{aligned} \tag{2.6}$$

where

$$g_h(z_t \mid c_i, x_t) = \frac{P(z_t \mid c_i = 1, x_t)}{P(z_t \mid c_i = -1, x_t)} \tag{2.7}$$

Note that $\lambda(c_i \mid z_{0:t-1}, x_{0:t-1}) = \log o(c_i \mid z_{0:t-1}, x_{0:t-1})$. The update rule can be thus simplified as

$$\begin{aligned}\lambda(c_i \mid z_{0:t}, x_{0:t}) &= \lambda(c_i \mid z_{0:t-1}, x_{0:t-1}) + \log\left(g_h(z_t \mid c_i, x_t)\right) \\ &= \lambda(c_i) + \sum_{s=0}^{t} \log\left(g_h(z_s \mid c_i, x_s)\right)\end{aligned} \tag{2.8}$$

$\lambda(c_i)$ is the prior of the map without any observations. The last equation indicates that the log odd ratio of a cell at a given time t is its accumulated sum from 0 to t.

During the inference time, the probability $\gamma_{i,t}$ can be derived from

$$\gamma_{i,t} = 1 - \frac{1}{1 + \exp\left(\lambda(c_i \mid z_{0:t}, x_{0:t})\right)} \tag{2.9}$$

We set the prior of the log-odd for each cell to be 0 with the assumption that initially a cell has equal likelihood of being occupied or free. A cell c_i is regarded as occupied at time t if $\lambda(c_i | z_{0:t}, x_{0:t})$ is greater than zero and free otherwise. When updating the map, we regard the LiDAR points as locations of obstacles and all the cells along the laser path as free. The rest of the cells will not be updated at time t.

We use the particle filter to localize robot's pose in the environment. The particle filter is a type of Bayes filters which approximates the probabilities in the prediction and update steps with a set of particles, namely a set of 3-vectors that represent the possible robot 2D positions

(x_t, y_t) and orientations θ_t in the world coordinate. Let's denote the number of particles as N. It is important to make N sufficiently large (the exact number is usually task-specific), as the particle filter is a sampling base probabilistic approximation. Initially, the robot has the equal likelihood of being at every particle, exemplified by the uniform weight $a_{p_i} = \frac{1}{N}$. In the prediction step, each particle will apply the IMU reading u_t plus a Gaussian noise $\beta_t = \mathcal{N}(\mu, \sigma^2)$ to estimate the robot's pose in $t+1$. Denote a pose estimation from one particle p_i as $x_{p_i,t}$, then

$$x_{p_i,t+1} = x_{p_i,t} + u_t + \beta_t \tag{2.10}$$

In the update step, we need to evaluate the goodness of our estimations with the new LiDAR scan z_{t+1}. A correlation score is calculated for each particle by matching the LiDAR scan z_{t+1} with the occupancy grid map constructed in the previous timestamp t, both of which have been transformed to the hypothesized particle frame. The higher the score is, the more similar the scan is with the environment. A simple correlation function could be

$$corr(z_{t+1}, m_t) = \sum_i \mathbf{1}(z_{t+1,i} = m_{t,i}) \tag{2.11}$$

We denote the LiDAR scan and the map in a particle p_i frame at time t as $z_t^{p_i}$ and $m_t^{p_i}$ respectively, but note that they have no difference with z_t and m_t from probability's perspective since the transformation is rigid and deterministic. We use the superscript to emphasize the reference frame that the variables are in.

All the correlation scores are converted into probabilities via softmax function. This probability represents the likelihood of the LiDAR scan given the predicted robot's pose and the environment.

$$P(z_{t+1}^{p_i} | x_{p_i,t+1}, m_t^{p_i}) = \frac{e^{corr(z_{t+1}^{p_i}, m_t^{p_i})}}{\sum_j e^{corr(z_{t+1}^{p_j}, m_t^{p_j})}} \tag{2.12}$$

The probability of $P\left(x_{p_i,t+1} | z_{t+1}^{p_i}, u_t\right)$ of each particle's estimation is the product of its weight

and $P(z_{t+1}^{p_i}|x_{p_i,t+1}, m_t^{p_i})$. We normalize them so that the total sum equals to one. In the end, the particle with highest probability is selected as the robot pose in time $t+1$ and the map is updated correspondingly.

After this step, $P\left(x_{p_i,t+1}|z_{t+1}^{p_i}, u_t\right)$ will be the new weight of particle p_i. To avoid the particle depletion [VDMDDFW00], we resample the particle distribution based on their sample importance when the inverse square sum of the weights is below a certain threshold $N_{threshold}$. After resampling, all the particles' weight will be set back to uniform distribution $a_{p_i} = \frac{1}{N}$ again.

In Figure 2.1, we compare the occupancy grid map generated from the particle filter and its counterpart generated from the dead-reckoning approach. The white area on the map is free space. The black area represents obstacles in the scene (e.g. wall). The gray area is the unknown area. We also illustrate the robot's trajectory as the red line on the map. Robot's pose estimation in the dead-reckoning approach is not corrected while we were building the map. Compared with the well-structured occupancy map using particle filter in the right, the map created from dead-reckoning in the left suffers from heavy distortion.

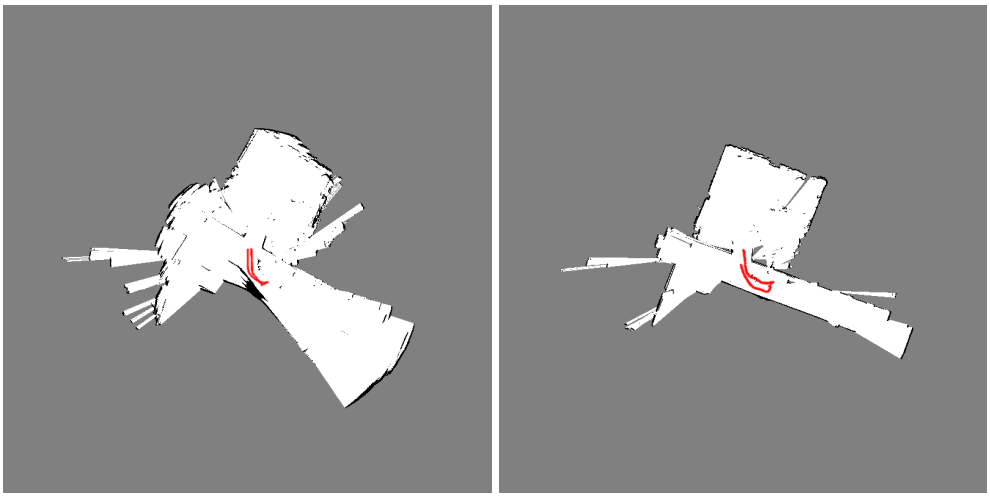


Figure 2.1: A comparison between the occupancy grid maps generated from dead-reckoning (left) and the particle filter (right) in the same environment. The white, black, and gray areas in the map correspond to free space, obstacles, and unknown areas of the scene. The robot's trajectory is shown as the red line in the image. There is an apparent distortion of the map in the left image, whereas the map on the right-hand side captures the correct layout of the scene.

2.2.3 Landmark-based Map

Instead of using a dense metric grid, the landmark-based map uses sparse landmarks to represent the environment. Here we present an example of landmark-based maps in an outdoor driving environment from KITTI dataset [GLSU13]. The dataset provides synchronized measurements from an IMU and a stereo camera. The intrinsic calibration matrix K and the baseline b of the stereo camera are known, so is the transform matrix ${}_C T_I \in SE(3)$ from IMU to the left camera. The landmarks are observed from the stereo camera and recorded in the pixel coordinates $z_t \in \mathbb{R}^{4\times N}$ where N is the number of detected landmarks at any time t. The association between 2D image features and the 3D landmarks is also known. For simplicity, we assume that all the landmarks are static.

The mapping process consists of two parts: landmark initialization and landmark update. The first part happens when we observe a new landmark in the 2D image space and need to infer its location in the 3D world space. Denote a landmark observed in the 2D image space as $z_t = (u_L, v_L, u_R, v_R)^\top$ where (u_L, v_L) is the pixel coordinates of the landmark in the left camera and (u_R, v_R) is the pixel coordinates of the landmark in the right camera. We can reconstruct the 3D location of the landmark $\mathbf{x}' = (x, y, z, 1)^\top$ in the camera frame by

$$z_t = M\frac{1}{z}\mathbf{x}' \tag{2.13}$$

where M is

$$M = \begin{bmatrix} fs_u & 0 & c_u & 0 \\ 0 & fs_v & c_v & 0 \\ fs_u & 0 & c_u & -fs_u b \\ 0 & fs_v & c_v & 0 \end{bmatrix} \tag{2.14}$$

b here is the baseline of the stereo camera and the rest of the parameters can be obtained from the intrinsic calibration matrix K. The uncertainty of the initialized landmark is assumed to be

Gaussian with the mean equals to the inhomogeneous point $\mathbf{x} = (x, y, z)^\top$ estimated above and the covariance equals to identity matrix I.

The landmark update step depends on the Extended Kalman Filter (EKF) to refine the estimations of the landmarks. Denote the pose of the IMU as T_t and its inverse as $U_t = (T_t)^{-1}$, we formulate the observation model of the system at time t as

$$z_t = h\left(U_t, m_t\right) + v_t = M\pi\left({}_C T_I U_t m_t'\right) + v_t \tag{2.15}$$

where z_t is the 2D observations in image space and m_t is the 3D landmarks. m_t' is the homogeneous representation of m_t. $\pi(\cdot)$ is a projection function for the camera:

$$\pi(\mathbf{q}) = \frac{1}{q_3}\mathbf{q} \in \mathbb{R}^{4\times N} \tag{2.16}$$

EKF assumes that both the prior of this observation model $P(m_t|z_{0:t}) \sim \mathcal{N}(\mu_t, \Sigma_t)$ and the observation noise $v_t \sim \mathcal{N}(0, V)$ are Gaussian distributions. Under this framework, we approximate the observation model with first-order Taylor series expansion:

$$z_t = M\pi({}_C T_I U_t \mu_t') + M\frac{\partial \pi}{\partial \mathbf{q}}({}_C T_I U_t \mu_t')_C T_I U_t P^\top \delta\mu_t + v_t \tag{2.17}$$

Here $P = [I|\mathbf{0}]$ is the standard projection matrix and μ_t' is the homogeneous representation of μ_t. The mean of predicted observations is

$$\bar{z}_t = M\pi\left({}_C T_I U_t \mu_t'\right) \tag{2.18}$$

and Jacobian of $\bar{z}_t$ with respect to m_t is

$$H_t = M\frac{\partial \pi}{\partial \mathbf{q}}({}_C T_I U_t \mu_t')_C T_I U_t P^T \tag{2.19}$$

which completes the total update rule of EKF,

$$\begin{aligned} K_t &= \Sigma_t H_t^\top \left(H_t \Sigma_t H_t^\top + I \otimes V \right) \\ \mu_{t+1} &= \mu_t + K_t \left(z_t - \bar{z}_t \right) \\ \Sigma_{t+1} &= \left(I - K_t H_t \right) \Sigma_t \end{aligned} \tag{2.20}$$

The operation $\otimes$ means creating a diagonal matrix based on V - the covariance of our observation noise. Landmarks that are not observed by the stereo camera will not be updated even though they have participated in the calculation of Kalma gain K_t.

The pose of the IMU is updated in the similar fashion. In the prediction, we assume that the prior $P(U_t|z_{0:t}, u_{0:t-1}) \sim \mathcal{N}(\mu_{t|t}, \Sigma_{t|t})$ is a Gaussian distribution with mean $\mu_{t|t} \in SE(3)$ and covariance $\Sigma_{t|t} \in \mathbb{R}^{6\times 6}$. The motion model of the robot is discretized into a time interval τ with the process noise $w_t \sim \mathcal{N}(0, W)$.

$$U_{t+1} = exp(-\tau(u_t \hat{+} w_t))U_t \tag{2.21}$$

The symbol $\hat{(\cdot)}$ is the skew symmetric metric for $se(3)$. There is a negative sign in $(u_t \hat{+} w_t)$ because U_t is the inverse IMU pose. u_t is the output of the IMU encoded in $se(3)$ space. Partitioning the motion model into nominal kinematics and perturbation kinematics can help us compute the mean and covariance of $P(U_{t+1}|z_{0:t}, u_{0:t}) \sim \mathcal{N}(\mu_{t+1|t}, \Sigma_{t+1|t})$ in the prediction step of EKF. The predicted observations based on $\mu_{t+1|t}$ is

$$\bar{z}_{t+1} = M\pi({}_C T_I \mu_{t+1|t} m_t) \tag{2.22}$$

and the Jacobian of $\bar{z}_{t+1}$ with respect to U_{t+1} is

$$H_{t+1|t} = M \frac{\partial \pi}{\partial \mathbf{q}} ({}_C T_I \mu_{t+1|t} m_j) {}_C T_I (\mu_{t+1|t} m_t)^\odot \tag{2.23}$$

where for each column $\mathbf{s} \in \mathbb{R}^3$ of $\mu_{t+1|t} m_t$, we have

$$\begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}^{\odot} = \begin{bmatrix} I & -\hat{\mathbf{s}} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 6} \tag{2.24}$$

and again $\hat{\mathbf{s}}$ is the skew symmetric metric of $\mathbf{s}$. In summary, the EKF update rule for IMU is

$$\begin{aligned} K_{t+1|t} &= \Sigma_{t+1|t} H_{t+1|t}^{\top} \left(H_{t+1|t} \Sigma_{t+1|t} H_{t+1|t}^{\top} + I \otimes V \right)^{-1} \\ \mu_{t+1|t+1} &= \exp\left(\left(K_{t+1|t} \left(z_{t+1} - \bar{z}_{t+1} \right) \right) \right) \mu_{t+1|t} \\ \Sigma_{t+1|t+1} &= \left(I - K_{t+1|t} H_{t+1|t} \right) \Sigma_{t+1|t} \end{aligned} \tag{2.25}$$

We jointly compute the Jacobian matrix H_t, innovation $z_{t+1} - \bar{z}_{t+1}$, and the Kalman gain K_t between the IMU and the landmarks to capture the correlation between them.

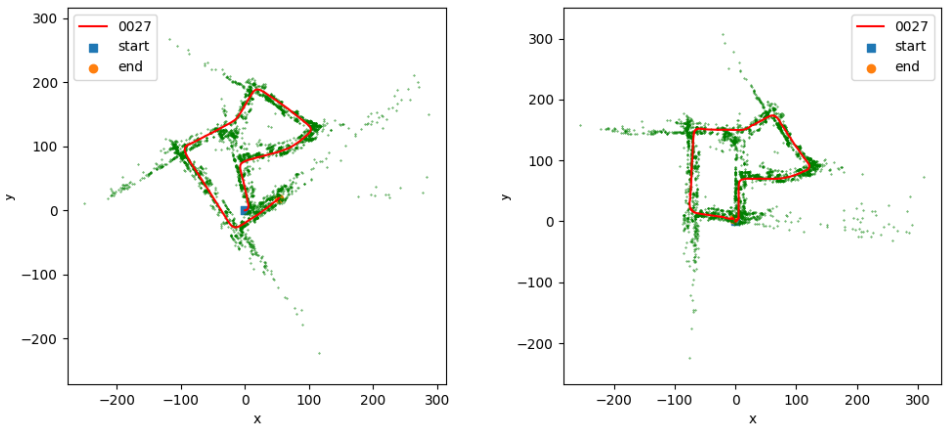


Figure 2.2: A comparison between the landmark-based maps generated from dead-reckoning (left) and the Extended Kalman filter (right) in the same environment. The green points are landmarks and the red line is the trajectory of the data collection vehicle. The vehicle was driving in a loop in an urban environment. No loop closure is performed for either algorithm, but the trajectory estimated from the EKF manages to detect the cyclic layout of the environment while the dead-reckoning approach fails.

The resulting two landmark-based maps from the dead-reckoning approach and the EKF are shown in Figure 2.2. The dead-reckoning approach estimates the landmark location without

any correction of the IMU pose estimation. The green points are the observed landmark location in the bird's eye view. The red line is the robot's trajectory. In the provided dataset, the data collection vehicle is supposed to stop at its initial position, creating a closed loop. However, the trajectory from dead-reckoning approach stops at a distance far away from the starting point whereas the trajectory generated from the EKF algorithm successfully detects the loops. The map generated from the EKF algorithm reflects a more structured street view where the data was collected.

2.3 Conclusion

In this chapter, we have discussed various kinds of map representation in robotics. We survey the literature and discuss the advantage and disadvantage of metric maps (e.g. occupancy grid maps, landmark-based maps), non-metric maps (e.g. topological map), and their combination (e.g. semantic maps). We then study two parametric mapping techniques that use Baye's filter to model the environment in SLAM problems. The occupancy grid map created by a particle filter algorithm and the landmark-based map created by Extended Kalman filter shows the power of this probabilistic framework that can significantly reduce maps' distortion and improve the robustness of mapping algorithms.

Chapter 3

Semantic Mapping

Examples given in the previous chapter have shown the parametric techniques to map the environments, but still the information they provide is not sufficient for the autonomous vehicle applications. Nowadays, many autonomous vehicles still depend on high-definition (HD) maps which offer centimeter-level definitions on the road, such as road networks, traffic signs, crosswalks, stop signs, traffic lights, and even speed limits. Today's construction of the HD maps still involves tedious manual labeling and many of them will become outdated during construction or road network changes. Given the fasting changing environments on the roads, obsolete HD maps will cause vehicles to perform inadequate reference for path tracking which may result in unsafe driving scenarios. In the process of HD map generation, extracting semantics and attributes from data takes the most amount of the work [Jia18]. A model that automates this process could improve HD map generation, reduce labor costs, and increase driving safety.

The retrieval of centimeter-level semantic labels of the scene is a non-trivial task. The conditional random field (CRF) framework is often adopted in the prior works [DFRDW11, SSLT12]. Recent advancements of deep learning also provide promising results in terms of retrieving semantic information from images. State of the art semantic segmentation algorithms such as [LSD15, ZSQ+17, CZP+18] generate pixel-level semantic labels with high pixel accuracy.

Researchers have also explored methods to create semantic maps of the environment: examples are given in [MCUS18, MLU17, HML+19]. Multi-sensor fusion has been used to improve the robustness of algorithms; however, these approaches either use aerial imagery to extract road information or do not explicitly map the lane and crosswalk information, which are required for HD maps. Therefore, a detailed semantic map for urban autonomous vehicle applications is still of interest to explore.

In this chapter, we are going to present a method that can automatically generate a dense probabilistic bird's eye view (BEV) semantic map in urban driving environments. Our work leverages the dense point maps built from a 16 channel LiDAR and a state-of-the-art semantic segmentation network to provide robust labels for roads, lane marks, crosswalks, and sidewalks in the map. The network is trained only on a publicly available dataset. The semantic map models the uncertainty of the semantic prediction with a confusion matrix formulation. In addition, we discover that the fusion of LiDAR intensity slightly improves the accuracy of land mark segmentation on the road. The comparison with a ground truth HD map that has been tested in our autonomous vehicle for campus mail delivery tasks shows that the proposed model can identify semantic features on the road and localize them accurately.

3.1 Background

Semantic segmentation is the task of assigning each observed data point (e.g. pixel or voxel) to a class label that contains semantic meanings. Research in this field has made tremendous progress as large scale datasets like CityScapes [COR+16], CamVid [BFC08], Mapillary[NORBK17], become available. Given that HD maps require fine-grained labeling for each object, building such maps can significantly benefit from the pixel level information provided by semantic segmentation algorithms.

In 2D semantic segmentation, predominant works [LSD15, ZSQ+17, CZP+18] leverage

pyramid like encoder-decoder architecture to capture both global and local information in the images. Trained on the large scale datasets aforementioned, these network architectures can easily detect objects on the road even when these objects only have textural differences like colors. In 3D semantic segmentation, work has also been done by modeling 3D LiDAR point clouds as range images and then feeding them to a classical CNN network to classify each point [WWYK18, HPSS20, WSY$^+$18]. While the results of these works seem promising, due to the nature of LiDAR sensors, these methods cannot distinguish objects with texture differences such as colors. Researchers have also proposed an alternative approach of segmentation on voxelized point clouds [TCA$^+$17]. However, such 3D convolutions are computationally expensive and usually require dense raw point cloud measurements (i.e 32 or 64-channel LiDARs), making it troublesome to operate in real time.

As we have mentioned in Chapter 2, by combining semantic labels with geometric information we can create a semantic map of the environment. Methods like [SSLT12] propose CRF based methods in dense semantic mapping. They use associative hierarchical CRF for semantic segmentation and pairwise CRF for mapping. The pairwise potential minimization enforces the output smoothness. Sengupta et. al. [SGST13b] utilize a stereo pair to estimate the depth robustly, but they do not explicitly map the lane and crosswalk information, which are required for HD maps. Daniel et. al. [MCUS18] fuse semantic images from a camera with LiDAR point clouds, but they use real-time raw point clouds from a 64-channel LiDAR providing much denser real-time information whereas we can build a map from a 16-channel LiDAR that has a lower cost. Additionally, their focus is on off-road terrains, in contrast to our focus on the urban driving scenario. Urban driving scenes require special treatment of classes that contain specific traffic rules, such as lane marks and crosswalks.

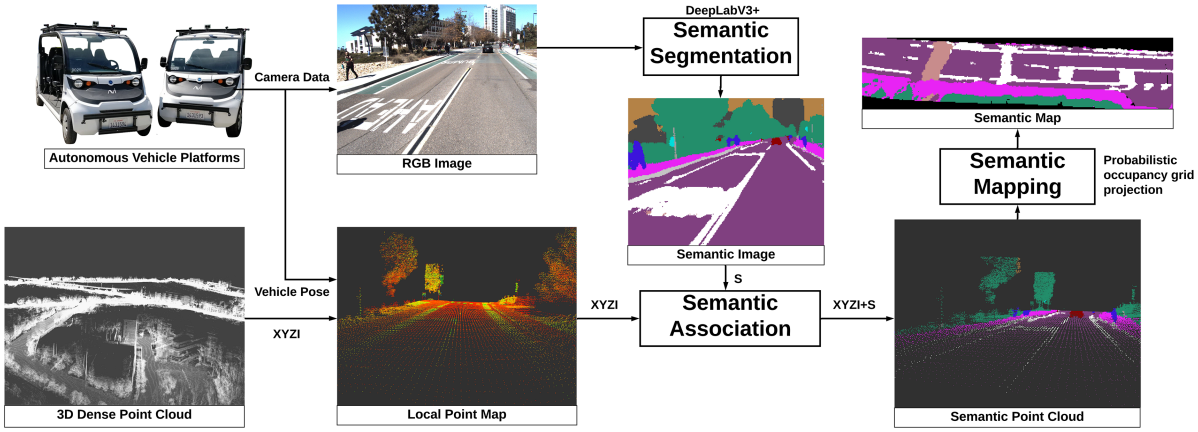


Figure 3.1: Our pipeline for generating probabilistic semantic maps for road feature extraction and HD mapping applications. Our model consists of three parts: 1) Semantic Segmentation, which predicts the semantic labels based on 2D images. 2) Semantic Association, which associates point clouds with predicted semantic labels. 3) Semantic Mapping, which captures the latent distribution of labels with a probabilistic semantic mapping method and augments the lane mark prediction with LiDAR intensity.

3.2 Proposed Method

Our model consists of three parts: semantic segmentation, semantic association, and semantic mapping. The overall architecture is shown in Figure 3.1. We first use a semantic segmentation network to predict semantic labels from 2D images and associate the semantic labels with densified 3D point clouds. We then utilize a probabilistic framework to estimate the distribution of labels on each cell of a bird's eye view grid map. In this section, we will describe each part in detail.

3.2.1 Image Semantic Segmentation

We use the DeepLabV3Plus [CZP+18] network architecture to extract the semantic segmentation from 2D images. Multilevel spatial pyramid CNN layers are used to capture both the global and local features of the scene. A skip connection is made from the encoder to pass low level image features to the decoder. A lightweight ResNeXt50 [XGD+17] pre-trained on

Table 3.1: Training Labels and Their Associated Colors for the Semantic Segmentation Network

Training Labels			
curb	crosswalk	road	sidewalk
building	person	bicyclist	motorcyclist
lane marking (white)	sky	vegetation	manhole
pole	traffic-sign	bicycle	bus
car	motorcycle	truck	

ImageNet [RDS+15] is used as our feature extraction backbone. Compared with other backbones like ResNet101 [HZRS16], it can achieve the same mean of intersection over union (mIoU) value with fewer parameters and faster inference times. We also adopt the depth-wise separable convolution inspired by [CZP+18, Cho17] in our spatial pyramid layers and decoder layers to improve the inference time further while preserving the same performance.

Our semantic segmentation network is trained in the Mapillary Vistas dataset [NORBK17]. This dataset provides a large number of pixel-level semantic segmented images with 66 different kinds of labels in autonomous vehicle scenarios. We reduce the labels into 19 classes by removing labels that are not essential in our driving environment (e.g. snow) and merging labels with similar semantic meanings together (e.g. zebra line and crosswalk). This decision is made based on the observation that some classes are unlikely to appear in our test environment. The details of label merging can be found in Section 3.3.1. All the training labels and their associated colors are presented in Table 3.1.

3.2.2 Point Cloud Semantic Association

Given a semantic image, estimating the relative depth for the semantic pixel data can help us reconstruct the 3D scene with semantic labels. This information, however, is usually not available. In Chapter 2.2.3 we have presented a method to estimate the depth information from stereo cameras, but its result depends on salient features which are prone to error on the road or in challenging lighting conditions. LiDAR sensors can readily capture the depth information of

the objects, but as they are often equipped with few optical channels (e.g. 16), it can be difficult to infer the underlying geometry in real time due to their sparse resolution. To alleviate this problem, our method leverages centimeter-level localization [PLH+19] to extract small dense regions of a previously built dense point cloud map. These smaller regions are then projected into the semantically segmented image to retrieve depth information. Creating such a dense point map can be automated and only requires driving through the area once and is thus less expensive than human labeling.

With localization in place, the intrinsic camera parameters and the relative transformation between the camera and LiDAR are used to project point cloud data into the 2D image space. The point cloud data is then associated semantically using the nearest neighbor search. The relative transformation between the camera and the LiDAR is estimated using the PnP method [LMNF09], and the intrinsic matrix of the camera is determined by a traditional chessboard method.

3.2.3 Semantic Mapping

While a point cloud with semantic labels naturally preserves the 3D geometry of the environment, such representation of the scene is subject to the sensor measurement noise and small semantic label fluctuations. To address this problem, we create a bird's eye view (BEV) semantic occupancy grid to model the environment. The semantic occupancy grid has height H, width W, and channels C. Each channel corresponds to a semantic class of the scene. When the semantic point cloud is constructed, we project it onto the grid using the x and y components of the point cloud. A point's semantic label will be regarded as the observed semantic label of its nearest cell c_{ij}. Every cell in the grid covers a $d \times d$ square area (in meters) of the physical world where the value d is a discretization factor.

Two types of maps are constructed with a focus on different purposes for the autonomous vehicle. One is a local probabilistic map in the ego-vehicle frame and another one is a global probabilistic map in the world frame. The former can provide direct dense semantic cues

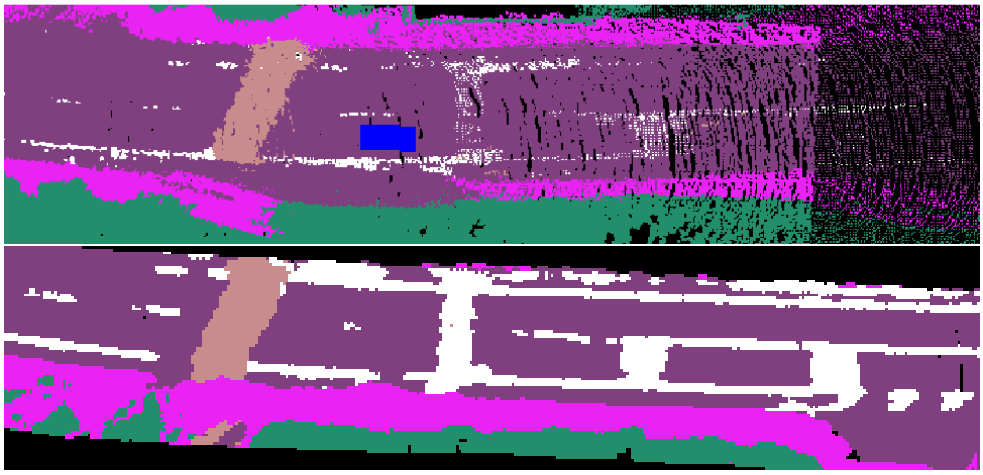


Figure 3.2: The map generation result of our method. The top image is the local map and the bottom image is the same region in the global map. We highlighted the ego-vehicle as a blue box on the local map.

for vehicle's navigation and controls whereas the latter can help the automation of HD map construction. Given a frame i where a local map is initiated, we update it continuously until the difference of the pose at i and new pose $i+t$ is beyond a threshold α. The new local map transforms the previous local map into its own frame and takes it as prior knowledge of the environment. A side-by-side visual comparison between these two maps is shown in Figure 3.2.

The robustness of the semantic occupancy grid estimation is enhanced by a probabilistic model which leverages both the semantic and LiDAR intensity information from the point cloud to reduce the prediction error. We denote the semantic label distribution across all the channels as $\mathbf{S}_t$, the observed semantic labels as $\mathbf{z}_t$, and observed LiDAR intensity as I_t. Hence, the task is to estimate $\mathbf{S}_t$ from our past observations, i.e. the probability distribution of $P(\mathbf{S}_t|\mathbf{z}_{1:t}, I_{1:t})$. By following the Markov assumption and assuming that observed semantic labels and LiDAR

intensity are conditionally independent given $\mathbf{S}_t$, the update rule of the semantic probability is

$$\begin{aligned} P(\mathbf{S}_t|\mathbf{z}_{1:t}, I_{1:t}) &= \frac{1}{\mathbf{Z}} P(\mathbf{z}_t, I_t|\mathbf{S}_t, \mathbf{z}_{1:t-1}, I_{1:t-1}) P(\mathbf{S}_t|\mathbf{z}_{1:t-1}, I_{1:t-1}) \\ &= \frac{1}{\mathbf{Z}} P(\mathbf{z}_t, I_t|\mathbf{S}_t) P(\mathbf{S}_t|\mathbf{z}_{1:t-1}, I_{1:t-1}) \\ &= \frac{1}{\mathbf{Z}} P(\mathbf{z}_t|\mathbf{S}_t) P(I_t|\mathbf{S}_t) P(\mathbf{S}_{t-1}|\mathbf{z}_{1:t-1}, I_{1:t-1}) \end{aligned} \tag{3.1}$$

where $\mathbf{Z}$ is a normalization factor. Here we also assume that

$$P(\mathbf{S}_t \mid \mathbf{z}_{1:t-1}, I_{1:t-1}) = P(\mathbf{S}_{t-1} \mid \mathbf{z}_{1:t-1}, I_{1:t-1}) \tag{3.2}$$

We model $P(\mathbf{z}_t|\mathbf{S}_t)$ with a 2D matrix $\mathbf{M}$ where an element in the ith row and jth column represents the likelihood of label i being predicted as label j. It characterizes our confidence of prediction to allow a more accurate probabilistic update. We model $P(I_t|\mathbf{S}_t)$ as a function of the reflectivity rate of each class in the scene. Inspired by the log-odd method in Section 2.2.2, value of these probability are calculated in the logarithmic space. This gives us an incremental update function and better numerical stability.

The intensity from LiDAR sensors serves a strong cue for different materials on the scene. For example, the top image in Figure 3.4 illustrates a BEV intensity map on a road segment. Because lane marks are painted white, they can reflect light at higher intensity and be segmented out with a threshold k. We can use this as a prior to reason about the layout of the scene: it can help for the cases when semantic segmentation fails to capture the true label in poor lighting conditions.

3.3 Experiments and Results

Our experimental data was collected by one of our experimental autonomous cars [PLH+19]. The car is equipped with a 16-channel LiDAR and six cameras. The cameras

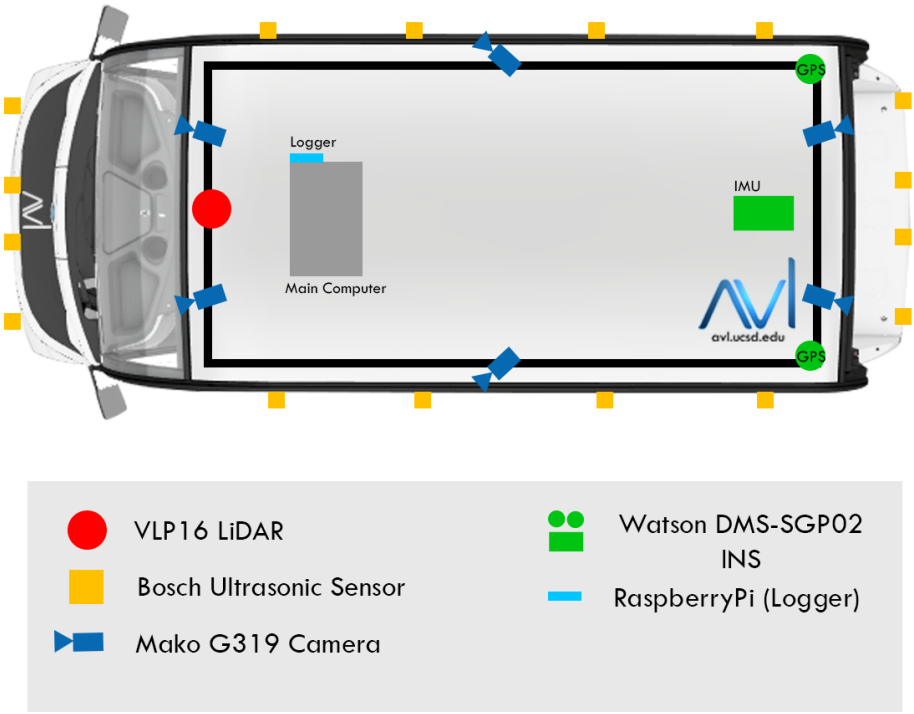


Figure 3.3: Data collection vehicle sensor configuration

are set up as two on the front, one on each side and two on the back as shown in Figure 3.3. Data from the front left camera, LiDAR, and vehicle position is recorded for experiments by driving along multiple areas at the UC San Diego campus. The camera data is streamed at approximately 13 Hz and the LiDAR scans at approximately 10 Hz. We drive through the campus to collect data for urban driving scenarios–including challenging scenarios like driving along steep hills, intersections and construction sites.

3.3.1 Image Semantic segmentation

Training Dataset

The Mapillary Vistas dataset is split into 18,000 training images and 2,000 validation images. We merge terrain into vegetation, different types of riders into the human category,

traffic-sign-back and traffic-sign-front into traffic-sign, bridge into building, and different kinds of crosswalks into a single crosswalk class. The training dataset is augmented by random horizontal flips with 0.5 probability, random resize with the scale ranging from 0.5 to 2, and random crop. These images are also normalized to a distribution with mean of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225). Due to the similarity of the Mapillary dataset and our driving scenarios, as well as the intense data augmentation in the training process, we do not visually observe severe performance drop when testing it in the UC San Diego campus.

Hyperparameters

We use a batch size of 16 with synchronized batch normalization [ZSQ+17] to train our network for 200 epochs on eight 2080Ti GPUs with input image sizes of 640x640. The output stride of the network is eight. We use a SGD optimizer and employ a polynomial learning rate policy [CZP+18, ZSR+19] where the learning rate is $base_lr \times (1 - \frac{epoch}{maxepoch})^{power}$ with 0.005 base learning rate and power=0.9. The momentum and weight decay are set to 0.9 and $4e^{-5}$, respectively.

Metric

The mean of Intersection over Union (mIoU) is used to evaluate the performance of the network. The mIoU of ResNeXt50 in the validation set is 68.32%. Compared with ResNet101, ResNeXt50's performance decreases slightly but requires much less memory (from 367MB to 210MB), which is preferable for our onboard hardware where the memory is limited. The inference time of the network is approximately 0.2s per image.

3.3.2 Semantic Mapping

We select a region that spans 1.1 km of the UC San Diego campus to evaluate our map generation results. An HD map has been manually annotated in this area. It contains road

Table 3.2: Quantitative evaluation on our labeled data for road, crosswalk and lane mark regions. Refer to section 3.3.2 for details.

Methods	IoU			mIoU	Accuracy		
	roads	crosswalks	lane marks		roads	crosswalks	lane marks
Vanilla	**0.715**	0.537	0.135	0.462	**0.797**	0.577	0.180
Vanilla+I	0.712	0.510	0.163	0.462	0.789	0.548	0.234
CFN	0.671	**0.605**	**0.301**	**0.526**	0.708	**0.696**	0.652
CFN+I	0.669	0.588	0.298	0.518	0.705	0.676	**0.657**

information such as crosswalks, sidewalks, and the center of road lane definitions and has been tested in realistic environments. Based on these labels, we generate our semantic map with five channels: *road, crosswalk, lane marks, vegetation, and sidewalk* with resolution $d = 0.2$ meters. The labeling of HD map is not a trivial work. However, it exactly showcases the value of being able to automate the entire process.

The mIoU and pixel accuracy are used as the metrics of our evaluation. While mIoU reflects our model's segmentation recall and precision, it is worth noting these metrics are influenced by the sparsity of the LiDAR point cloud: the model output may be very accurate but may contain a lot of unclassified cells (holes). We address this problem by using a smoothing kernel to interpolate the missing labels on our map.

For all the experimental results shown in Table 3.2, we clip the local dense point maps extracted up to 15m along the longitudinal axis and -10 to 10 m along the lateral axis of the vehicle as the semantic segmentation performance drops significantly for a longer range. The quantitative comparisons are performed in the global frame.

Modeling of Observation Uncertainty

We first validate the design of matrix $\mathbf{M}$ that models the observation uncertainty of the semantic label $\mathbf{z}_t$ given $\mathbf{S}_t$. The elements of $\mathbf{M}$ is recorded logarithmically for numerical stability. Two approaches have been investigated: one is defined by $\mu(\mathbf{I} + \lambda\mathbf{1})$ where λ is a hyper-parameter and μ is a normalization factor: we refer to this model as Vanilla. Another approach is the

confusion matrix of the semantic segmentation network in the Mapillary validation dataset: we define this as CFN. During the inference time, we assign each cell to the label with the highest probability. Quantitative results are shown in the first and third row of Table 3.2. Compared with the Vanilla version, CFN has significant improvements in IoU and pixel accuracy on the classes of crosswalks and lane marks. This indicates the advantage of utilizing the confusion matrix of the network to model the prediction error in the semantic segmentation and thus provides a better map generation result.

Integration with LiDAR Intensity

In order to utilize the fact that different materials on the road have different reflectivity, we first remove all the intensity data that is less than $k = 14$; this normalized threshold was manually calibrated for a Velodyne VLP-16 LiDAR. An example of this filtered point cloud is shown in Figure 3.4. Then during the semantic mapping step, if a label is predicted as lane marks, we increase its logarithmic probability by a constant factor γ. This essentially suppresses our prediction of other classes and increases our prediction confidence for lane marks. The models that contain the integration of intensity have a "+I" in Table 3.2. For Vanilla+I, we observe better results with respect to Vanilla in terms of accuracy and IoU of the lane marks but slightly decreases for roads and crosswalks, indicating the benefit of intensity integration for lane mark prediction. This tendency, however, does not replicate for CFN+I with respect to CFN. To achieve further improvement, a more sophisticated function may be needed to model the LiDAR intensity.

An example of the global map generated by our CFN+I model for the entire test region is shown in Figure 3.5. A region of the map has been amplified in the figure, showing that our model can capture the static elements on the road clearly.

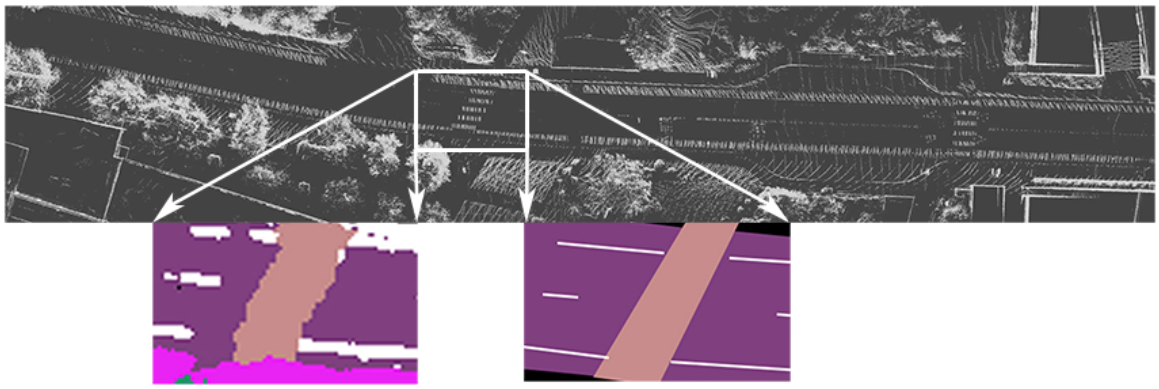


Figure 3.4: A visualization of our generated map (bottom left), the ground truth label (bottom right), and the LiDAR point cloud map (top). The point cloud map has been thresholded based on their intensity value. Any points that have intensity below a certain threshold k are discarded. With this threshold, we can clearly see the layout of the crosswalk and lane marks on the road.

3.3.3 Comparison to Sparse LiDAR Scan

One possible alternative for associating semantic images with depth information is to use the point cloud data generated by the LiDAR in real time. By following a similar mapping approach, we project the point cloud onto the semantic image frame and build the semantic map. Figure 3.6 demonstrates the real-time performance of this approach in the local frame. For the 16-channel LiDAR used, the point cloud scans are too sparse to construct a semantic map at greater distances. This becomes worse when the car drives faster. Therefore, a pre-built dense point cloud map allows us to construct semantic maps for longer ranges.

3.3.4 Comparison to Planar Assumption

Another method we explored is back-projecting the 2D semantic image into the 3D space using a homography with the assumption that the ground is flat. The back-projection approach leaves no black holes. Nevertheless, the planar assumption fails along steep hills or road intersections in urban driving scenarios as shown in Figure 3.7: this leads to considerable distortion at longer ranges.

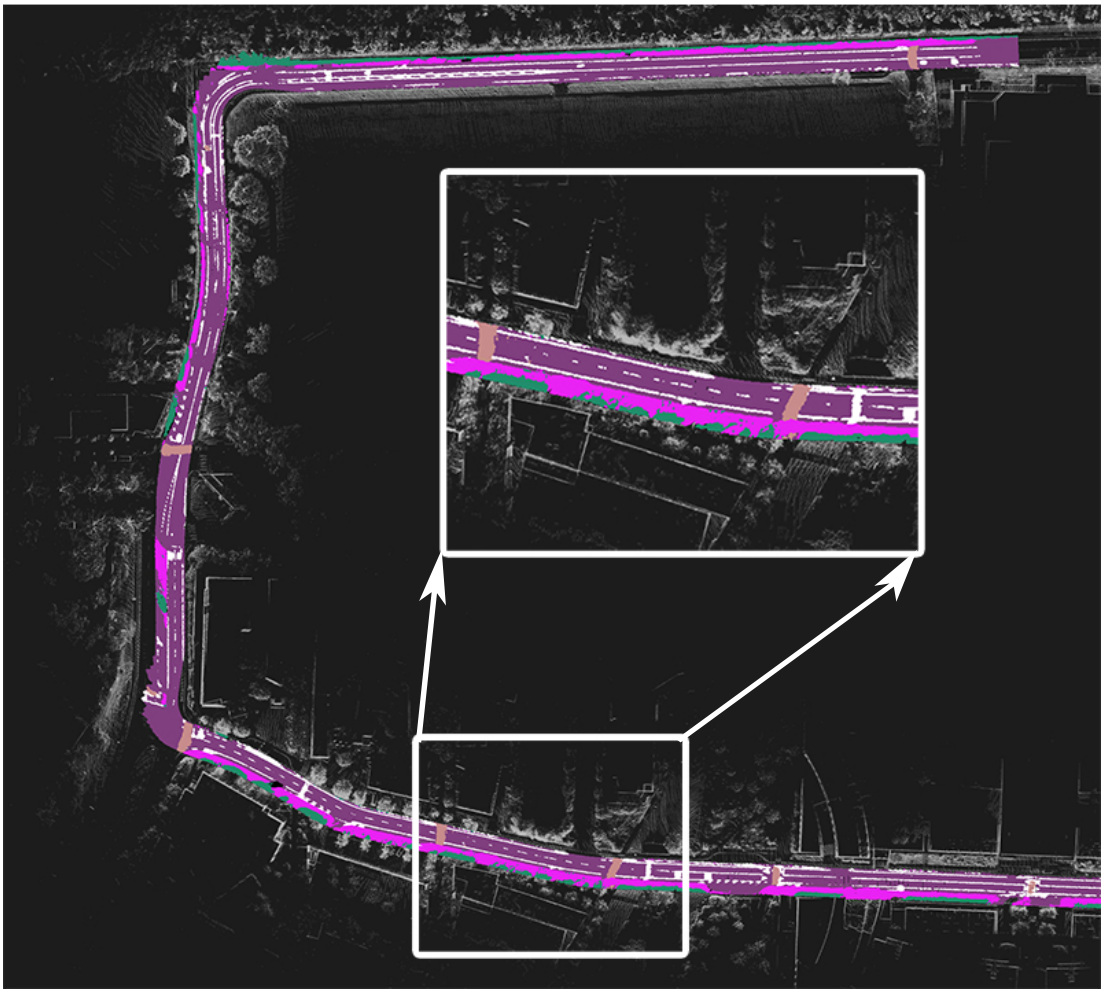


Figure 3.5: The BEV version of our generated map in the entire region of our testing dataset. Our semantic map is displayed on top of the dense point cloud map. One segment of the map has been magnified to show the details of the map. Best view in color.

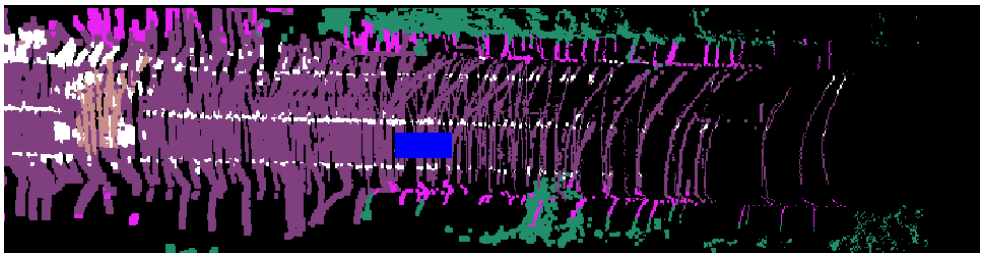


Figure 3.6: A semantic map generated from real-time LiDAR scan in the local frame. The black region on the scene indicates the unknown area which is not detected by the LiDAR sensor. The ego-vehicle is highlighted as a blue box on the map.

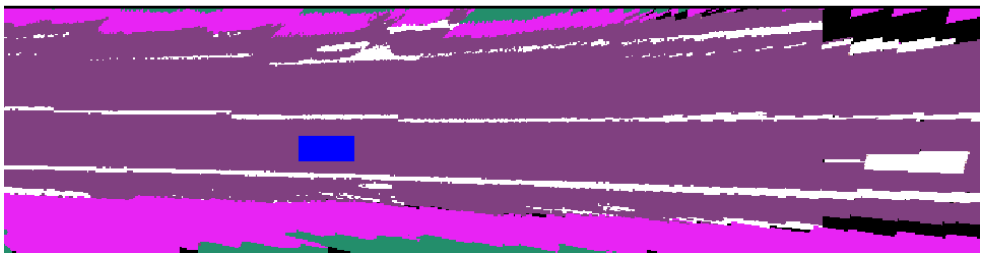


Figure 3.7: A local semantic map generated by back-projecting 2D semantic image into the 3D space with planar assumption. The map presented in the vehicle frame. The distortion of lines indicates the failure case. The ego-vehicle is highlighted as a blue box on the map.

3.4 Conclusion

In this chapter, we introduce a statistical mapping method which fuses the rich information from 2D semantic images with 3D point clouds and creates a semantic map for autonomous vehicle applications. Our comparisons to manually annotated maps indicate the efficiency of this work on identifying road features and localizing them in a bird's eye view. While the current HD maps involve massive manual laboring, this method can be viewed as a step toward the automation of HD map annotation for crosswalks, lane markings, drivable surfaces, and sidewalks. Semantic features from our maps can be incorporated into the generation of HD maps independently of predefined HD map formats and further extended for center lane identifications which are often used for path tracking algorithms.

3.5 Acknowledgments

Chapter 3 contains material as it appears in the International Conference on Intelligent Robots and Systems 2020. Paz, David; Zhang, Hengyuan; Li, Qinru; Xiang, Hao; Christensen, Henrik I., The thesis author was one of the primary investigators and authors of this paper.

Chapter 4

Conclusion and Future Directions

In this thesis, we study the mapping problem for autonomous vehicle applications. We have surveyed the existing literature for different kinds of map representations. The parametric mapping examples in Chapter 2 have shown the advantage of using probabilistic to handle the uncertainty of the environment and the measurement noise. We present two different models of Bayes' filter mapping techniques that create a dense and sparse metric map respectively. In Chapter 3, we propose a new method to automatically generate a bird's eye view semantic map in urban driving environments. Our approach leverages the LiDAR sensor and a deep learning based semantic segmentation network to provide robust labels for roads, land marks, crosswalks, and sidewalks in the map. The entire mapping process involves little human intervention, paving the way to the fully-automated HD map generation for autonomous vehicles.

By accounting for the road network junctions and forks, a future direction of our work is the full automation of road network annotations that could leverage graphical methods. While a combination of the mapping techniques proposed can potentially address the scalability drawbacks from HD maps, they also point out new areas of research on high-level dynamic planning. Currently, many autonomous driving architectures require dense point cloud maps for localization and come at a scalability and maintenance cost in a similar way that HD maps do. By dynamically

estimating drivable surfaces, traffic lanes, lane markings, and other road features, the notion of using centimeter-level localization could be removed as long as immediate actions can be extracted from a high-level planner. In future work, we plan to seek out solutions for fully automating the HD mapping process while exploring the idea of dynamic planning without a detailed dense point cloud map.

Bibliography

[AC03] Philipp Althaus and Henrik I. Christensen. Automatic map acquisition for navigation in domestic environments. *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2:1551–1556 vol.2, 2003.

[BFC08] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, xx(x):xx–xx, 2008.

[CCC+16] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309?1332, Dec 2016.

[Cha12] Yu-Wei Chao. Semantic structure from motion with points, regions, and objects. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR ?12, page 2703?2710, USA, 2012. IEEE Computer Society.

[Cho17] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[CN08] Mark Joseph Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27:647 – 665, 2008.

[CN11] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[COR+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[CPSA17] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[CZP+18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *Lecture Notes in Computer Science*, page 833?851, 2018.

[DFRDW11] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte. Classification and semantic mapping of urban environments. *The International Journal of Robotics Research*, 30(1):5–32, 2011.

[DRMS07] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[DTK05] A. Diosi, G. Taylor, and L. Kleeman. Interactive slam using laser and advanced sonar. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1103–1108, 2005.

[ED09] Ethan Eade and Tom Drummond. Edge landmarks in monocular slam. *Image Vision Comput.*, 27(5):588?596, April 2009.

[Elf89] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46?57, June 1989.

[FS00] Elisabetta Fabrizi and Alessandro Saffiotti. Extracting topology-based maps from gridmaps. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 3:2972–2978 vol.3, 2000.

[FWDT05] Patrick S. Foo, William H. Warren, Andrew P. Duchon, and Michael J. Tarr. Do humans integrate routes into a cognitive map? map- versus landmark-based navigation of novel shortcuts. *Journal of experimental psychology. Learning, memory, and cognition*, 31 2:195–215, 2005.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2014.

[GFLM17] Saurabh Gupta, David Fouhey, Sergey Levine, and Jitendra Malik. Unifying map and landmark based representations for visual navigation, 2017.

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[GM98] S. Gillner and H. A. Mallot. Navigation and acquisition of spatial knowledge in a virtual maze. *Journal of Cognitive Neuroscience*, 10(4):445–463, July 1998.

[HML+19] Namdar Homayounfar, Wei-Chiu Ma, Justin Liang, Xinyu Wu, Jack Fan, and Raquel Urtasun. Dagmapper: Learning to map by discovering lane topology. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2911–2920, 2019.

[HPSS20] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. Cnn-based lidar point cloud de-noising in adverse weather. *IEEE Robotics and Automation Letters*, 5(2):2514–2521, 2020.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[Jia18] J. Jiao. Machine learning assisted high-definition map creation. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 01, pages 367–373, July 2018.

[KA08] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[KG15] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 04 2015.

[Kui00] Benjamin Kuipers. The spatial semantic hierarchy. *Artif. Intell.*, 119(1?2):191?233, May 2000.

[LBH15] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[LL07] T. Lemaire and S. Lacroix. Monocular-vision based slam using line segments. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2791–2796, 2007.

[LMNF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81, 02 2009.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[LSN+16] Stephanie M. Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David D. Cox, Peter I. Corke, and Michael Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32:1–19, 2016.

[MAMT15] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147?1163, Oct 2015.

[MAT17] Raul Mur-Artal and Juan D. Tardos. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255?1262, Oct 2017.

[MCUS18] Daniel Maturana, Po-Wei Chou, Masashi Uenoyama, and Sebastian Scherer. Real-time semantic mapping for autonomous off-road navigation. In Marco Hutter and Roland Siegwart, editors, *Field and Service Robotics*, pages 335–350, Cham, 2018. Springer International Publishing.

[ME85] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 116–121, 1985.

[MLU17] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3438–3446, 2017.

[MM94] H. Moravec and M. J. Martin. Robot navigation by 3d spatial evidence grids. 1994.

[Mor88] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9:61–74, 1988.

[NORBK17] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *International Conference on Computer Vision (ICCV)*, 2017.

[NPB95] Illah R. Nourbakhsh, Rob Powers, and Stanley T. Birchfield. Dervish - an office-navigating robot. *AI Magazine*, 16:53–60, 1995.

[PJ12] A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *2012 IEEE International Conference on Robotics and Automation*, pages 3515–3522, 2012.

[PLH+19] David Paz, Po-Jung Lai, Sumukha Harish, Hengyuan Zhang, Nathan Chan, Chun Hu, Sumit Binnani, and Henrik Christensen. Lessons learned from deploying autonomous vehicles at UC San Diego. In *Field and Service Robotics*, Tokyo, JP, August 2019.

[PNGC19] Anwesan Pal, Carlos Nieto-Granda, and Henrik I. Christensen. Deduce: Diverse scene detection methods in unseen challenging environments. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.

[PS17] Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning, 2017.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[Ros58] Frank F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[SAH04] Chanop Silpa-Anan and Richard I. Hartley. Localisation using an image-map. 2004.

[SGST13a] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr. Urban 3d semantic modelling using stereo vision. In *2013 IEEE International Conference on Robotics and Automation*, pages 580–585, 2013.

[SGST13b] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr. Urban 3d semantic modelling using stereo vision. In *2013 IEEE International Conference on Robotics and Automation*, pages 580–585, May 2013.

[SK02] H. Shatkay and L. P. Kaelbling. Learning geometrically-constrained hidden markov models for robot navigation: Bridging the topological-geometrical gap. *Journal of Artificial Intelligence Research*, 16:167?207, Mar 2002.

[SLL02] Stephen Se, David Lowe, and J.J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research*, 21:735–760, 08 2002.

[SLL05] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

[SSLT12] S. Sengupta, P. Sturgess, L. Ladick, and P. H. S. Torr. Automatic dense visual semantic mapping from street-level imagery. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 857–862, Oct 2012.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, Cambridge, Mass., 2005.

[TC06] E. A. Topp and H. I. Christensen. Topological modelling for human augmented mapping. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2257–2263, 2006.

[TCA+17] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. *2017 International Conference on 3D Vision (3DV)*, Oct 2017.

[Thr98] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artif. Intell.*, 99:21–71, 1998.

[Thr02] Sebastian Thrun. Robotic Mapping: A Survey. *Science*, 298(5594):699f–699, oct 2002.

[TINGC10] A Trevor, Rogers III, Carlos Nieto-Granda, and Henrik Christensen. Tables, counters, and shelves: Semantic mapping of surfaces in 3d. 10 2010.

[VDMDDFW00] Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric Wan. The unscented particle filter. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS?00, page 563?569, Cambridge, MA, USA, 2000. MIT Press.

[WGGH18] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[WS02] Ranxiao Frances Wang and Elizabeth S. Spelke. Human spatial representation: Insights from animals. *Trends in Cognitive Sciences*, 6(9):376–382, September 2002.

[WS08] D. F. Wolf and G. S. Sukhatme. Semantic mapping using mobile robots. *IEEE Transactions on Robotics*, 24(2):245–258, April 2008.

[WSY+18] Yuan Wang, Tianyue Shi, Peng Yun, Lei Tai, and Ming Liu. Pointseg: Real-time semantic segmentation based on 3d lidar point cloud, 2018.

[WWYK18] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[XGD+17] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[YLL+18] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graphgeneration. *Lecture Notes in Computer Science*, page 690?706, 2018.

[YML05] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 903–910 Vol. 1, 2005.

[YMV+14] S. Yi, S. McGill, L. Vadakedathu, Q. He, I. Ha, J. Han, H. Song, M. Rouleau, D. Hong, and D. D. Lee. Thor-op humanoid robot for darpa robotics challenge trials 2013. In *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 359–363, 2014.

[ZSQ+17] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[ZSR+19] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.