An Information Theoretic Framework for Camera and Lidar Sensor Data Fusion and its Applications in Autonomous Navigation of Vehicles

by

Gaurav Pandey

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering–Systems)
in The University of Michigan
2014

Doctoral Committee:

Associate Professor Ryan M. Eustice, Co-Chair
Associate Professor Silvio Savarese, Co-Chair
Professor Alfred O. Hero III
Assistant Professor Honglak Lee

ACKNOWLEDGEMENTS

It has been five years when I boarded that flight to DTW (a completely different world) to follow my dreams of getting a PhD. Since then it's been a journey full of learning and I have tried to soak in as much knowledge as I could. As I stand at the end of this road, I would like to thank those who are an integral part of this beautiful journey.

First of all I would like to thank my supervisor Ryan Eustice for his excellent guidance and support throughout my graduate studies. I really appreciate his patience to discuss and explain even the most trivial questions that I have asked him several times. He always had time to meet and discuss any problem. He always made sure that I am making progress and always guided me towards the right path. Thanks Ryan for always keeping me on track by your formula of *"Must, Should and Would be nice"*. I will remember this mantra of success for the rest of my life.

Next I would like to thank my co-advisor Silvio Savarese for his invaluable inputs on my research. Those weekly meetings helped me to stay focused and steadily progress towards my goal. I would also like to thank my committee members Alfred Hero and Honglak Lee for their valuable suggestions at times.

I will like to thank Jim McBride for his continuous support and guidance through these years. He has always been very helpful in providing resources needed to carry out the research presented in this thesis. I would like to thank him for those endless hours of data collection from that F-250 pickup truck.

I will also like to thank the gang in PeRL lab: Ayoung, Nick, Jeff, Paul, Steve and Ryan, without them it would have been really hard to complete this journey. I will never forget the interesting discussions on various topics (research/non-research) that we had in the lab and during the lunch sessions at commons cafe. I also want to take this opportunity to thank my room-mate Stephen for all those discussions that we had about other important stuff in life, besides research.

Finally, I would like to thank my family who supported me unconditionally from the time I decided to leave my country for higher education. My mother (Bharti Pandey) and father (Mahesh Chandra Pandey) have always been a great source of motivation for me. I thank them for always standing by my side in success as well as in failures. It is because of

them that I was always able to deal with failures and never gave up. I also thank my brother Saurabh for his support throughout these years. Lastly, I would like to thank my beloved Shikha for the time and support she provided to help me achieve this milestone.

Funding

This work is supported through grants from the Ford Motor Company via the Ford-UofM Alliance (Awards #N008265, #N009933 and #N015392).

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

This thesis develops an information theoretic framework for multi-modal sensor data fusion for robust autonomous navigation of vehicles. In particular we focus on the registration of 3D lidar and camera data, which are commonly used perception sensors in mobile robotics. This thesis presents a framework that allows the fusion of the two modalities, and uses this fused information to enhance state-of-the-art registration algorithms used in robotics applications. It is important to note that the time-aligned discrete signals (3D points and their reflectivity from lidar, and pixel location and color from camera) are generated by sampling the same physical scene, but in a different manner. Thus, although these signals look quite different at a high level (2D image from a camera looks entirely different than a 3D point cloud of the same scene from a lidar), since they are generated from the same physical scene, they are statistically dependent upon each other at the signal level. This thesis exploits this statistical dependence in an information theoretic framework to solve some of the common problems encountered in autonomous navigation tasks such as sensor calibration, scan registration and place recognition. In a general sense we consider these perception sensors as a source of information (i.e., sensor data), and the statistical dependence of this information (obtained from different modalities) is used to solve problems related to multi-modal sensor data registration.

CHAPTER I

Introduction

Today, robots are used to perform challenging tasks that were not possible twenty years ago because of limited computational and sensor resources. In order to perform these complex tasks, robots need to sense and understand the environment around them. Depending upon the task at hand, robots are often equipped with different sensors to perceive their environment. Two important categories of perception sensors mounted on a robotic platform are:

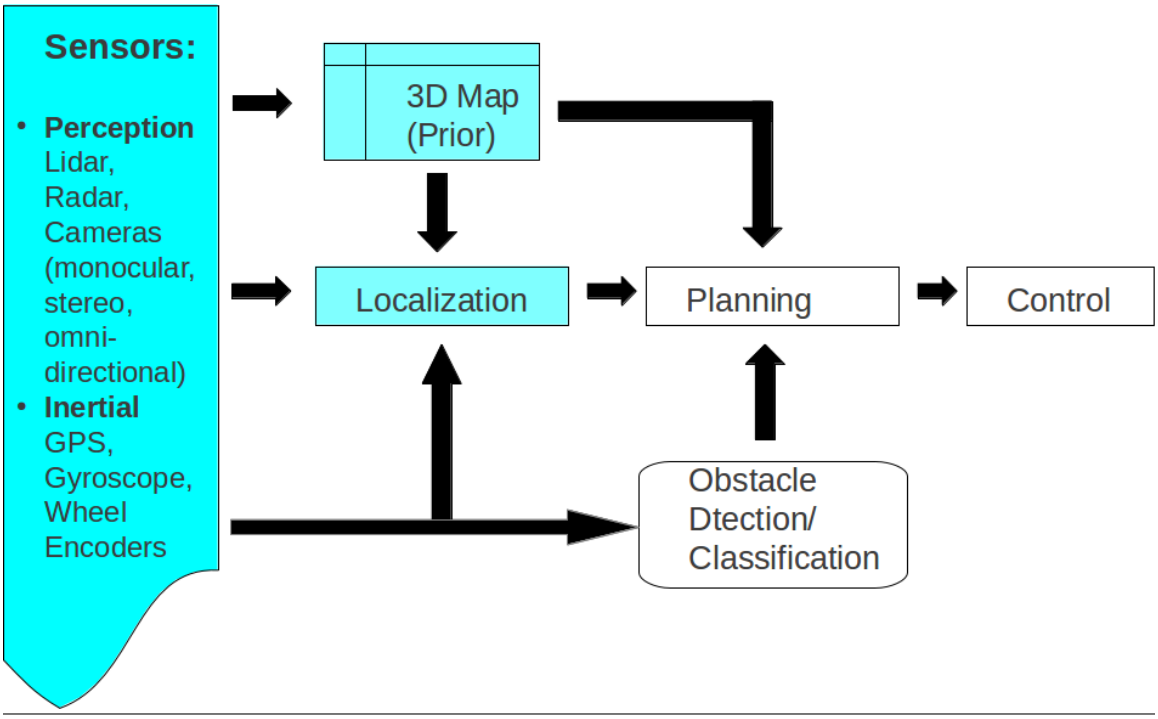
- **Range sensors**– 3D/2D lidars, radars, sonars, etc.
- **Cameras**– perspective, stereo, omnidirectional, etc.

With the recent advancements in these sensing technologies, the capabilities of robots to perform difficult tasks has been greatly extended. One such example was seen in the 2007 DARPA Urban Grand Challenge [87, 101, 111]. In this competition the robots had to automatically navigate through a $96\ \mathrm{km}$ urban area course in less than six hours. They had to follow all the traffic rules and even negotiate the intersections by interacting with other robots. The competition was a great success with four cars finishing the course in time. The University of Michigan and Ford Motor Company were one of the finalists in the competition and have continued to collaborate on autonomous ground vehicle research since the 2007 DARPA Urban Grand Challenge. Through this continued collaboration we have developed an autonomous ground vehicle testbed based upon a modified Ford F-250 pickup truck (Fig. 1.1). In the 2007 DARPA Urban Grand Challenge this vehicle showed capabilities to navigate in the mock urban environment, which included moving targets, intermittently blocked pathways, and regions of denied global positioning system (GPS) reception [101]. During the 2007 DARPA Urban Challenge the data obtained from the vehicle's perception sensors was mostly used independently, despite the fact that these sensors capture complementary information about the environment. The 3D point cloud

Figure 1.1 A collage of the modified Ford F-250 truck used in DARPA Grand Urban Challenge showing the sensor configuration and the computers. Sensors are strategically placed around the vehicle. The lidar and omnidirectional camera are mounted on top so that an entire 360 degree view of the environment is captured.

captured by the Velodyne laser scanner gives entirely different information about a scene as compared to the image of the same scene captured by an optical camera system; however, the underlying structure generating the two signals (3D point cloud / image) is the same. Thus, the two signals are statistically dependent upon each other (i.e., knowledge of one tells us something about the other). We believe that, if using these sensors independently allows the robot to automatically navigate through complex urban environments, then fusing the data obtained from these sensors can greatly enhance the robustness of various state-of-the-art robotics algorithms. It is not new to fuse multi-modal data by exploiting their statistical dependence. In fact, registration of multi-modal data by maximizing the Mutual Information (MI) has been state-of-the-art in the medical imaging community for over two decades [98, 132, 158, 166]. However, these techniques are not popular among the robotics community despite the fact that robots today are often equipped with multiple sensing modalities. Therefore, through the work presented in this thesis we intend to encourage researchers to explore the possibilities of utilizing information theoretic concepts to increase the robustness of sensor-data registration algorithms required for autonomous navigation of vehicles. Here we present an information theoretic framework for fusion of data obtained from perception sensors mounted on a robotic platform, thereby opening the doors for a new inter-disciplinary research.

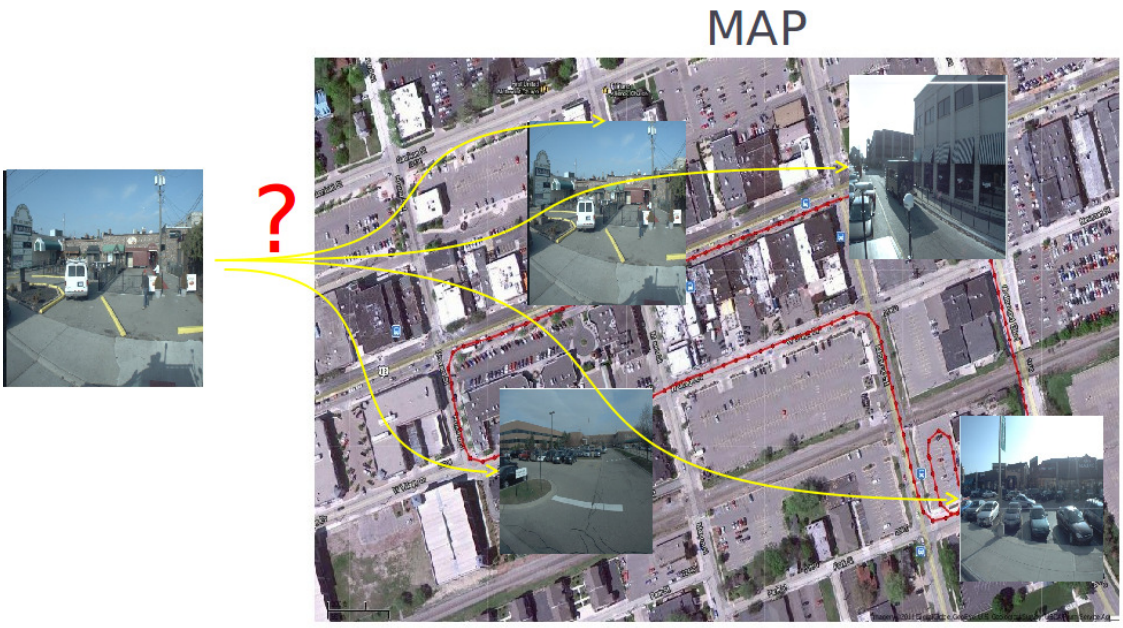
Figure 1.2 Block diagram showing different processes used in autonomous navigation of a vehicle.



1.1 Problem Statement

In this thesis we will present an information theoretic framework for registration of multi-modal data obtained from a 3D lidar and camera system mounted on an autonomous vehicle platform. The utility of precise registration of such multi-modal data is that it allows for robust automatic navigation of mobile robots. A typical autonomous navigation system is complex, and consists of several different modules as shown in Fig. 1.2. In this thesis we are mainly concerned about what the robot *sees* from the different perception sensors, and how we can extract useful information necessary for robust automatic navigation from these multi-modality sensors. In order to automatically navigate through an environment that has been mapped *a priori*, the first thing that the robot needs to do is to localize itself in the map (Fig. 1.3). Localization of robot can be achieved from GPS information, but in the absence of GPS or any other inertial sensor, the robot needs to register the current sensor data (lidar/camera) with the data in the prior map. It is typical to generate map data with vehicles equipped with highly precise inertial systems and then use that map data for localization [22, 27]. In static environments, the registration of current sensor data with the map data is fairly easy. However, in real-world applications, the environment is generally dynamic and the data captured on different days can appear significantly different. Therefore, the task of recognizing a location in the prior map becomes extremely challenging (Fig. 1.4).

Figure 1.3 Illustration of the place recognition problem. The robot makes a sensor observation and recognizes the location by registering the current sensor observation with the sensor data present in the map.

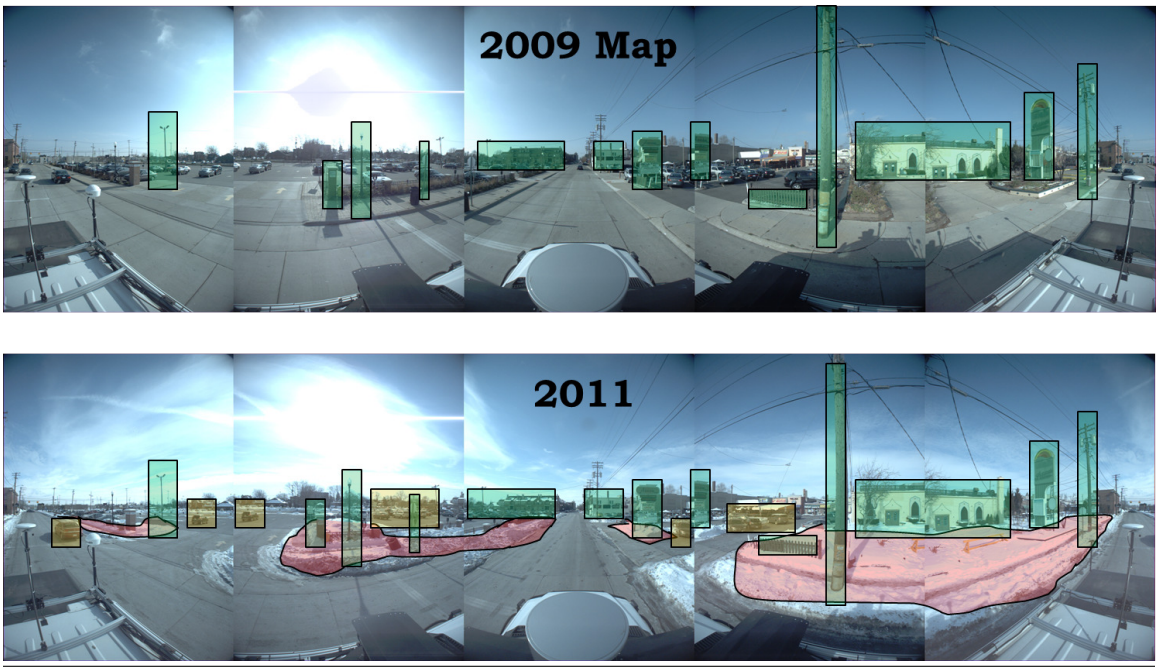


The drastic changes in the environmental appearance due to changing seasons, lighting conditions, and dynamical objects makes the task of localization extremely difficult. The image of a location captured on a bright summer day can appear significantly different from the image of the same location captured on a gray snow-covered winter day. So, if we use the image data alone to localize on a snowy winter day within an *a priori* map collected on a sunny summer day, then it might not be possible. However, if we use data from lidar as well, then we can boost the registration process by using the complementary information provided by the different sensing modalities. Thus, here we provide a framework for fusing the information obtained from these multi-modal sensors in a statistical framework and use them to enhance the robustness of the algorithms required for autonomous navigation of vehicles.

1.2 Research Approach

In this thesis we present a comprehensive analysis of fusion of data obtained from different modality perception sensors mounted on a robotic platform. In particular, we focus on a 3D lidar and camera system mounted on an autonomous vehicle testbed. We show that the robustness of registration algorithms used for autonomous navigation of vehicles can be greatly enhanced by fusing the multi-modal data obtained from these perception

Figure 1.4 The top panel shows the omnidirectional image of a location captured in fall 2009. The bottom panel shows the omnidirectional image of the same location in winter 2011. The significant change in the scene is clearly visible from the two images, for example, snow on the ground (marked in red), dynamic objects (marked in orange), lighting conditions, etc. Such drastic changes make registration of the 2009 and 2011 datasets a challenging problem. However, there are also common objects (marked in green) that have stationary statistics and can be used for registration of sensor data.



sensors. In this thesis we use a three-step approach, where we first co-register the lidar and camera data within a single scan by extrinsically calibrating the sensors (Chapter III). Extrinsic calibration of sensors allows reprojection of the 3D points from the reference frame of the point cloud to the image plane. This allows us to associate image pixel information with individual points in the point cloud. In the second step we align two such textured 3D point clouds, comprised of co-registered lidar and camera data, captured sequentially (Chapter IV). Here we assume that the overlapping regions of these sequential scans are mostly unchanged except for dynamic objects, which appear as noise in the overall registration process. The alignment of sequential scans provides an estimate of the vehicle ego-motion, which is used to create accurate 3D maps of the environment within a simultaneous localization and mapping (SLAM) framework. In the third and final step we use this textured 3D map of the environment and localize the robot within this prior map in an information theoretic framework (Chapter V).

1.3 List of Publications

Most of the work described in this thesis has either been published or submitted for review in peer-reviewed robotics conferences and journals. A list of publications resulting from the work presented in this thesis is given below:

1.3.1 Journal

1. Gaurav Pandey, James R. McBride and Ryan M. Eustice, *Ford campus vision and lidar data set*. International Journal of Robotics Research, 30(13):1543–1552, November 2011.

2. Gaurav Pandey, James McBride, Silvio Savarese and Ryan Eustice, *Automatic Targetless Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information*. Journal of Field Robotics, Submitted, Under Review.

1.3.2 Peer-Reviewed Conference Proceedings

1. Gaurav Pandey, James McBride, Silvio Savarese and Ryan Eustice, *Extrinsic calibration of a 3d laser scanner and an omnidirectional camera*. In 7th IFAC Symposium on Intelligent Autonomous Vehicles, pages 336–341, Leece, Italy, September 2010.

2. Gaurav Pandey, James R. McBride, Silvio Savarese and Ryan M. Eustice, *Visually bootstrapped generalized ICP*. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2660-2667, Shanghai, China, May 2011.

3. Gaurav Pandey, James McBride, Silvio Savarese and Ryan Eustice, *Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information*. Special Track on Robotics in AAAI Conference, pages 2053–2059, Toronto, Canada, July 2012.

4. Gaurav Pandey, James McBride, Silvio Savarese and Ryan Eustice, *Toward Mutual Information based Automatic Registration of 3D Point Clouds*. IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2698–2704, Vilamoura, Portugal, October 2012.

5. Gaurav Pandey, James McBride, Silvio Savarese and Ryan Eustice, *Toward Mutual Information Based Place Recognition*. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, May 2014 (Accepted, To Appear).

1.4 Thesis Overview

The remainder of this thesis is organized as follows:

- **Chapter II** contains a review of probability and information theory as it relates to concepts exploited in this thesis. In this chapter we explain all the key concepts used as building blocks for developing the information theoretic framework for sensor data fusion and its applications in autonomous navigation of vehicles.

- **Chapter III** describes the first step necessary for fusing the data from camera and lidar modalities, that is, extrinsic calibration of the two sensors. In this chapter we describe two techniques for calibrating the sensors: *(i)* target-based and *(ii)* target-less. The target-less method maximizes the mutual information between the camera and lidar data to estimate the calibration parameters in an information theoretic framework. Quantitative and qualitative comparison of the output of these methods show that the proposed target-less calibration method is more robust and easy to use in practical in-field operations.

- **Chapter IV** describes the second step of our research approach, where we align two successive 3D scans comprised of co-registered lidar data and camera imagery. Here we present two methods of scan alignment, one that uses the camera and lidar modalities to align the scans in a decoupled way, and another that combines them in an information theoretic framework.

- **Chapter V** describes the final place recognition algorithm, where we present a robust information theoretic framework for localizing within an *a priori* map, without any inertial sensor or GPS prior. We show that using data from different sensing modalities enhances the robustness of the algorithm.

- **Chapter VI** presents a summary of the work presented in this thesis and provides some concluding remarks.

CHAPTER II

Probability and Information Theory

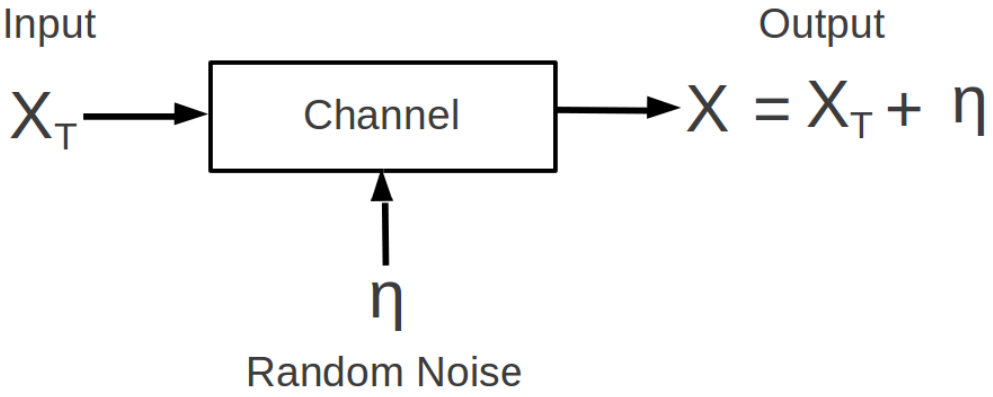
One of the main contributions of this thesis is to apply the concepts from information theory into robotics applications. Information theory is a branch of applied mathematics that deals with quantification of information present in a random variable. Claud E. Shannon [149], often attributed as *"the father of information theory"*, introduced these concepts and used them to find the fundamental limits on compression of digital signals in a communication system. Since its introduction, the concepts from information theory have been successfully used in statistical inference [1, 116, 144], bioinformatics [54, 24], medical imaging [74, 132, 157, 166], computer vision [71, 76, 162] and many other disciplines of science that involve random variables at the core. Any physical problem, modeled as a set of random variables derived from an underlying probability distribution (known or unknown), can be analyzed and solved using concepts from information theory.

In this chapter we review the concepts of probability and information theory that will be used in subsequent chapters. We will also introduce the notations used throughout the thesis to represent random variables and other information theoretic quantities like entropy, mutual information, etc. This chapter is mainly intended as a review of the probability and information theory concepts used in this thesis. We will explain all the key concepts relevant to the work done in this thesis. However, we recommend the readers to refer to any of the standard probability [48, 56, 138] and information theory [130, 29] textbooks for more details. Readers familiar with these concepts may easily skip this chapter.

2.1 Random Variables and Probability Distribution

Real-world systems are generally uncertain in nature and probability theory is a tool that helps in the modeling and estimation of this uncertainty. It is impossible to build a perfect system without any uncertainties. However, we can model and estimate the uncertainties

Figure 2.1 Signal flow through a noisy communication channel. The input signal is corrupted by channel noise resulting into a random output.



in any real-world system using concepts from probability theory. In any real-world system the measurements or observations that take numerical values which are not certain and change over repeated trials are considered to be *random variables*. These observations can be a signal passing through a noisy communication channel (Fig. 2.1), a noisy range measurement from a laser scanner, or an image captured from a noisy camera, etc. The randomness introduced in the measurements due to the noise in the system can be modeled in several different ways. Generally it is considered to be additive with certain known characteristics or probability distribution. Therefore the output of a noisy system can be written as:

$$X = X_T + \eta, \tag{2.1}$$

where X is the noisy observation, X_T is the true value, and η is the random noise added to the observation. The random noise causes the observations to be uncertain and therefore it can be modeled as a random variable. Depending upon the values this random variable takes, it is classified as a discrete or continuous random variable. If a random variable takes only certain distinct real values then it is called a discrete random variable. For instance, if we consider the outcome of a roll of a fare dice to be a random variable X, then X can take only six different values $[1, 2, 3, 4, 5, 6]$. Whereas if a random variable can take any real number within a given interval then it is called a continuous random variable. For example, if we consider the x coordinate of the point where the dart hits the target in a dart game (assuming the center of the target to be the origin) to be a random variable X, then $X \in [-R, R]$, where R is the radius of the circular target. In this thesis we consider only discrete random variables although most of the theory presented here can be easily extended to continuous random variables also. For most parts of this thesis when we say a random variable we mean a discrete random variable unless specified otherwise.

A discrete random variable is charecterized by its *probability mass function*, which defines the probability that the random variable X is exactly equal to some value x:

$$p_X(x) = Probability([X = x]). \tag{2.2}$$

Here $[X = x]$ defines an event for which the random observation X of a probabilistic experiment is mapped onto a real number x. If there does not exist an event for which an observation X is mapped onto x then $[X = x]$ is called a *null* event and the probability of occurrence of that event is 0. Since $p_X(x)$ is a probability it should satisfy:

$$0 \leq p_X(x_i) \leq 1 \text{ and } \sum_{x_i} p_X(x_i) = 1, x_i \in \Re. \tag{2.3}$$

We will often use the notation $p(x)$ instead of $p_X(x)$ to denote the probability mass function of a random variable X.

2.2 Uncertainty and Entropy

Entropy is a measure of the amount of uncertainty in a random variable. It was first used by Shannon [149] to quantify the expected value of the information contained in a message passing through a noisy communication channel. He used it to provide some fundamental limits on the lossless encoding of digital signals in a communication system, assuming that the signal can be represented as a sequence of independent and identically distributed (IID) random variables. Shannon's entropy for a discrete random variable X that takes on values $[x_1, x_2, \cdots, x_n]$ with the probability $p(x_i)$ $(i = \{1, 2, \cdots, n\})$ is given by:

$$H(X) = -\sum_{i}^{n} p(x_i) \log_b p(x_i). \tag{2.4}$$

The entropy of X can also be interpreted as the expected value of the random variable $\log_b \frac{1}{p(x)}$:

$$H(X) = E_X[\log_b \frac{1}{p(x)}], \tag{2.5}$$

where E_X is the expectation or the average value with respect to random variable X,

$$E_X[g(X)] = \sum_{x_i} g(x_i)p(x_i), \tag{2.6}$$

where $g(X)$ is any function of random variable X.

The value of entropy depends upon the base of the logarithm (b) used in (2.4). The most common base of logarithm used to calculate entropy is 2 and the entropy calculated with $\log_2$ is expressed in $bits$. However, we can use any base b to calculate the entropy. If entropy is calculated with $\log_e$ then it is expressed in $nats$. It should be noted that the base of the logarithm in (2.4) only scales the value of entropy by a scalar because of the following property of logarithm:

$$\log_b p = \log_b a \log_a p. \tag{2.7}$$

Therefore, if $H_a(X)$ and $H_b(X)$ are the entropies calculated with base a and b, respectively, then they are related as:

$$H_a(X) = \log_a b H_b(X). \tag{2.8}$$

Thus, the base of the logarithm is just a scaling factor and can be chosen based on the data. In this work we will consider the base of logarithm to be 2 (since that is the standard) unless specified otherwise.

Entropy has some desirable properties as far as quantification of uncertainty of a random variable is concerned. One of the most important properties of entropy is that it is a function of the probability distribution of the random variable and is therefore independent of the actual values that the random variable takes. Variance $(E_X[(X - E_X[X])^2])$ of the random variable is another way to quantify uncertainty but it is dependent upon the values that the random variable takes and is therefore misleading sometimes. For instance, if we have a random variable X that takes on two values $\{0, 1\}$ with probability $\{p(0) = 0.5, p(1) = 0.5\}$, then the variance of this random variable is 0.25 and its entropy is 1. Now consider another random variable Y that takes values $\{0, 10\}$ with same probabilities $\{p(0) = 0.5, p(10) = 0.5\}$ the variance now changes to 25 whereas the entropy of this random variable is still 1. Statistically, random variables X and Y are similar and have the same amount of uncertainty, which is correctly depicted by the entropy function. This property of entropy proved extremely useful in the work presented in this thesis. In the real-world, certain physical quantities are measured by different techniques resulting in observations that do not necessarily map to the same values. Using entropy as a measure of their uncertainties allows us to analyze these observations in a pure statistical sense (free from the actual values of the observations) as the underlying probability distribution of these observations is essentially the same. Moreover, entropy is a concave function of the probability distribution of the random variable [29], a property that we will use in subsequent chapters.

2.3 Statistical Dependence and Mutual Information

In section 2.2 we described entropy of a single random variable, i.e. the amount of uncertainty in the random variable. However, when there are more than one random variable in a system, we need to observe them simultaneously and analyze the uncertainty of the combined system. If the random variables constituting the system are statistically dependent, then the observation of one random variable affects the probability of observing the other. Whereas, if the random variables X and Y are independent, then the observations of X does not inform the distribution of Y, and the joint distribution of X and Y factors into the product of their marginals:

$$p_{XY}(x, y) = p_X(x)p_Y(y). \tag{2.9}$$

Mutual Information (MI) is one of the most popular measures that quantifies this statistical dependence of the random variables. It is expressed as the Kullback-Leibler (KL) divergence [81] of the product of the marginal distributions $(p(x)p(y))$ and the joint distribution $(p(x, y))$ of the random variables:

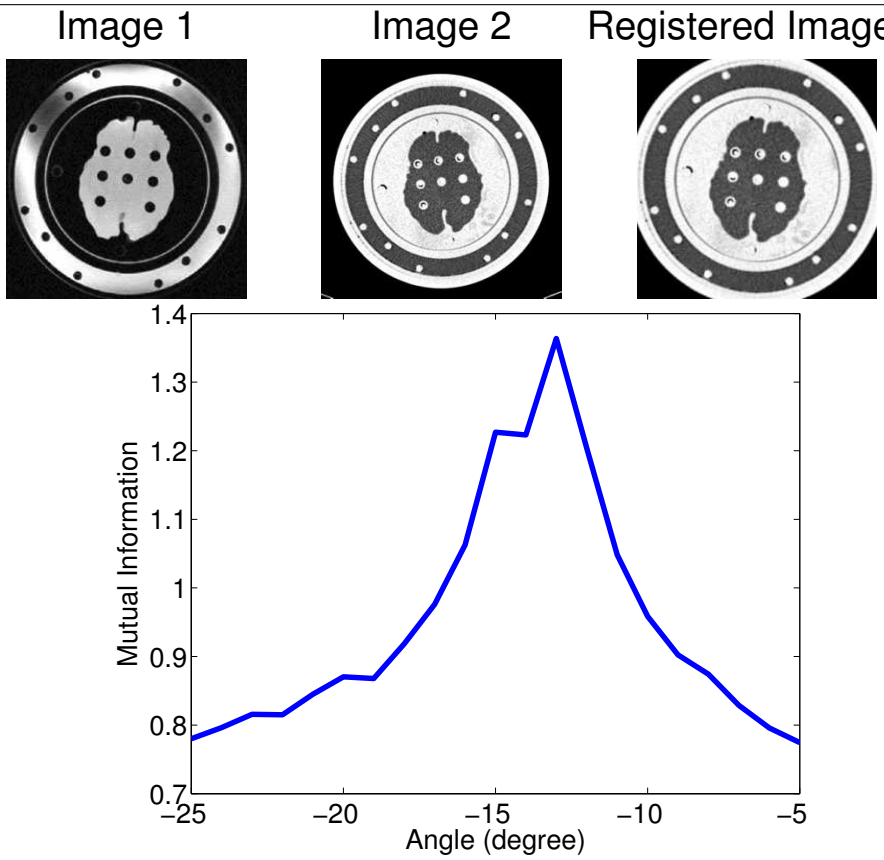
$$\mathrm{MI}(X, Y) = \sum_{i=0}^{n} \sum_{j=0}^{m} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)}. \tag{2.10}$$

Therefore, if X and Y are independent, then MI equals 0 because $p(x, y) = p(x)p(y)$ in that case. Alternatively, MI can also be expressed in terms of entropies of the random variables:

$$\begin{aligned} \mathrm{MI}(X, Y) &= H(X) - H(X|Y) && (2.11)\\ &= H(Y) - H(Y|X) && (2.12)\\ &= H(X) + H(Y) - H(X, Y). && (2.13) \end{aligned}$$

These relationships between MI and entropy are derived from the KL divergence given in (2.10). The proofs for these relationships are provided in appendix A. Intuitively, MI is the amount of information that X contains about Y and vice versa. The relationship between MI and entropy of the random variables (2.13) shows that $\mathrm{MI}(X, Y)$ is the reduction in the amount of uncertainty of the random variable X when we have some knowledge about Y. If X and Y are independent, then the knowledge of X does not provide any additional information about Y and hence the value of MI is 0. We can easily obtain this result by substituting the independence condition (i.e., $p(x, y) = p(x)p(y)$) in (2.10). However, if X

Figure 2.2 Multimodal image registration by maximization of mutual information [8]. The MI-based objective function maximizes at the correct value of rotation.



and Y are identical then X contains all the information of Y and vice versa. Therefore, the mutual information is the same as the uncertainty of either X or Y. Substituting $X = Y$ in (2.13) we get:

$$\mathrm{MI}(X, X) = H(X) + H(X) - H(X, X) = H(X) + H(X) - H(X) = H(X). \quad (2.14)$$

Thus, MI of a random variable with itself is equal to the entropy of the random variable. The entropy is therefore also called the self-information of the random variable.

In the subsequent chapters of this thesis we focus on registration of data obtained from two different sensors, specifically a 3D lidar and an optical camera system. We will be extensively using MI as a measure of statistical dependence of data obtained from these different sensing modalities.

MI-based registration dates back to the early 1990s when Woods et. al. [169, 170] first introduced a registration method for multi-modal images. His method was based on the assumption that images of the same object taken from different sensors (e.g., Magnetic

Resonance Imaging (MRI) [55], Positron Emission Tomography (PET) [10], Computed Tomography (CT) [67]) have similar gray values. In a more ideal case the ratio of gray values of corresponding points in a particular region of the image should have a very low variation. Thus, they proposed a method to minimize the average variance of this ratio in order to obtain the registration parameters. Hill et al. [60] extended this idea and constructed a joint histogram of the gray values of the two images and showed that the dispersion of the histogram is minimum when the two images are aligned. Although these ideas by Woods and Hill gave new insight into this problem, they still lacked a strong mathematical intuition. This mathematical intuition was provided by Collignon et al. [28] and Studholme et al. [157] when they suggested to use entropy as the measure of registration. Once entropy was introduced as a measure for registration of multi modality images, people started using more concepts from information theory. Soon thereafter, Viola and Wells [166] and Maes et al. [98] almost simultaneously introduced the idea of mutual information and provided rigorous mathematical background to the registration problem of multi modality images (Fig. 2.2). A year later Studholme et al. [158] introduced Normalized Mutual Information (NMI) and showed that it is more robust. Pluim et al. [131] introduced a multi-resolution approach to rigid registration of medical images (MRI, PET, CT) based on MI and NMI. The multi-resolution approach was aimed to accelerate the registration process while maintaining the accuracy and robustness of the method. The MI based techniques of multi-modal image registration showed great promise and very soon became a popular measure in medical image alignment. A comprehensive survey of mutual information based techniques up through 2002 is provided by Pluim et al. [132]. These techniques are widely used in many clinical applications for both rigid and non-rigid medical image registration. Klein et al. [74] presented a comprehensive evaluation of different optimization techniques used in non-rigid multi-modality image registration based on mutual information and B-Splines.

Numerous variations of the MI-based registration have been proposed to increase the accuracy, and robustness of the method. Knops et al. [75] proposed a NMI based registration using k-means clustering and shading correction. Shams et al. [147] showed that conventional mutual information based registration using pixel intensities is computationally expensive and ignores spatial information. Instead, they introduced the concept of gradient intensity as a measure of spatial strength of an image in a given direction. They estimate the transformation parameters by maximizing the mutual information between gradient intensity histograms. Luan et al. [97] proposed a novel quantitative-qualitative measure of mutual information (Q-MI) for multi-modality image registration. They argue that the conventional information measures, e.g., Shannon‘s entropy and mutual informa-

tion, reflect quantitative aspects of information because they only consider probabilities of events. They proposed to incorporate the utility/saliency of the event as well in the measure of information. Thus, they proposed the novel Q-MI measure in which the utility of each voxel in an image is determined according to the regional saliency value calculated from the scale-space map of the image. They showed that the Q-MI based registration method is more robust, compared to conventional MI-based registration methods, because it provides a smoother registration function with a relatively larger capture range. Staring et al. [155] proposed a graph based implementation of α-mutual information (α-MI). They also derived an analytical derivative of α-MI and used a stochastic gradient descent method to solve the registration problem.

The algorithmic developments in mutual information based registration problem were exponential during the late 1990s and early 2000s and very soon became state-of-the-art in the medical image registration field. Researchers widely used the MI-based algorithms focusing on specific problems in various clinical applications. Although, these mutual information based registration algorithms are widely used, they require significant CPU time for calculating and optimizing the mutual information. Lin and Medioni [94] showed that the mutual information computations can be fully parallelized and can be efficiently ported onto the GPU architecture. Recently Shams et al. [148] presented a real time implementation of mutual information based 3D medical image registration on a GPU. Besides medical image alignment, MI-based registration has also been used to solve problems involving a network of multi-modal sensors for environment monitoring and surveillance. Ertin et al. [33] proposed a MI-based dynamic sensor selection method for distributed tracking as well as stationary target localization using acoustic arrays. Krotosky and Trivedi [80] proposed a mutual information based registration of multi modal stereo videos for person tracking. They demonstrated successful registration of objects in color and thermal imagery and evaluated the algorithm in scenes with multiple objects at different depths and levels of occlusions. A good review of the information theoretic approaches for sensor management (multi-target tracking and classification) in a network of multi-modal sensors is given in Hero et al. [59].

Despite significant advancements in the MI-based registration techniques, they have not been widely used by the robotics community. Even though robots today are often equipped with multi-modal sensors to sense and understand the environment around them, it is not common to use information theoretic measures to analyze the data obtain from these different sensing modalities. In this thesis, we have used concepts from information theory to solve some common problems in robotics, thereby hoping that more researchers will explore this area of research.

2.4 Estimation of Entropy and Mutual Information

Estimation of entropy and MI from the observed data is a challenging task and has been extensively researched in the past. Since MI can be written as a function of marginal and joint entropies of the random variables under consideration, we can estimate MI from the entropy directly (2.13). It is common to estimate the entropy first and then use it to calculate an estimate of the MI. In this section we will mainly discuss some commonly used entropy estimators that can easily be used to calculate an estimate of MI.

2.4.1 Maximum Likelihood Estimator

One of the simplest and most commonly used estimators of entropy is the maximum likelihood (ML) estimator and is constructed by plugging-in the ML estimate of the probability mass function (PMF) of the random variable into (2.4). Let us consider a discrete random variable X that can take values in $\mathcal{X} = \{a_1, a_2, \cdots, a_d\}$ with certain unknown PMF $\{p(x = a_k) = X_k; k = [1, 2, \cdots, d]\}$. Here we use the notations common in information theory and call the realizations of random variable X, *words*, and the number of possible realizations of X the *vocabulary* size. If $\{x_1, x_2, \cdots, x_d\}$ are the observed counts of these *words* from the experiment then the ML estimate of the PMF is given by:

$$\hat{p}^{ML}(x = a_k) = \hat{X}_k^{ML} = \frac{x_k}{n}, \tag{2.15}$$

where $n = \sum_{k=1}^{k=d} x_k$ is the total number of observations. Substituting the ML estimate of the PMF from (2.15) into (2.4) we get the ML estimate of entropy of random variable X. It is important to note that although the ML estimate of the PMF $(\hat{X}_k^{ML})$ is unbiased, the corresponding plug-in estimate of the entropy exhibits substantial bias. Several methods for bias correction of the ML estimate has been proposed in the past [106, 127]. These methods estimate the bias and subtract it from the ML estimate of entropy. The bias-corrected estimators perform well when the observations are large but the sample size needed for good estimates increases quickly with the vocabulary size d of the random variable [68, 51]. Therefore, when $n \gg d$ the ML estimate provides robust and optimal estimates. However, when $n \ll d$, i.e., when the number of observations are much less as compared to the *vocabulary* size (common in practical applications), the ML estimate has high mean-squared-error (MSE) and significantly underestimates the true entropy. The James-Stein (JS) estimator provides significant improvement on the MSE of the ML estimator. This method was proposed by Hausser and Strimmer [54] for entropy and MI estimation, and is based on shrinking the ML estimator of the distribution of a random variable X toward a

target distribution $\mathrm{T} = [\mathrm{T}_1, \mathrm{T}_2, \cdots, \mathrm{T}_d]$:

$$\hat{X}_k^{JS} = \lambda \mathrm{T}_k + (1-\lambda)\hat{X}_k^{ML}, \tag{2.16}$$

where $\hat{X}_k = p_X(x = a_k)$ and $\lambda \in [0, 1]$ is a shrinkage coefficient used to optimize the estimation of MI. The choice of the target distribution is application specific and needs to be identified empirically but the optimal shrinkage coefficient λ can be estimated from the data by maximizing a quadratic risk function. Hausser and Strimmer [54] showed that this estimator works well in the under-sampled regime (i.e., $n \ll d$), where the number of observations are significantly less as compared to the vocabulary size. James-Stein estimators perform well in the under-sampling regimes when the vocabulary size is known. However, if the vocabulary size is unknown we need to account for the missing *words*. The coverage-adjusted estimator [24] is specifically designed to calculate an optimal estimate of entropy when there are missing *words* in the observations. In this approach the entropy of the random variable (with few observations, $n \ll d$) is estimated by applying the Horvitz-Thompson estimator [65] in combination with the Good-Turing correction [125] of the maximum likelihood estimate (MLE). The Good-Turing-corrected probability estimates are given by:

$$\hat{X}_k^{GT} = (1 - \frac{m_1}{n})\hat{X}_k^{ML}, \tag{2.17}$$

where m_1 is the number of bins with single observation (i.e., $x_k = 1$ and $\hat{X}_k^{ML}$ is the ML estimate). Combining this with the Horvitz-Thompson estimator, the required entropy is:

$$\hat{H}^{CS}(X) = -\sum_{k=1}^{n} \frac{\hat{X}_k^{GT} \log(\hat{X}_k^{GT})}{\left(1 - (1 - \hat{X}_k^{GT})^n\right)}. \tag{2.18}$$

2.4.2 Bayesian Estimator

Bayesian estimators regularize the ML estimates of the PMF using a certain prior and uses the resulting posterior distribution to estimate entropy. A Dirichlet prior with a fixed parameter α has been widely used to estimate entropy of random variables with known and finite vocabulary size. The posterior distribution for a Dirichlet prior with a fixed parameter $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_d\}$ is given by [95, 43]:

$$\hat{p}(x = a_k | \alpha_k) = \hat{X}_k^B = \frac{x_k + \alpha_k}{n + \xi}, \tag{2.19}$$

where $\xi = \sum_{k=1}^{k=d} \alpha_k$. Several variations of the parameter α has been proposed in the past to decrease the bias in the Bayesian estimator of entropy [1, 63, 79, 144]. However,

Nemenman et al. [116] showed that Bayesian estimators based on these priors are very sensitive to the sample size and become extremely biased when the number of observations are small (i.e., $n \ll d$). In order to fix this issue he proposed a Dirichlet-mixture prior that significantly improves the performance of the estimator but increases the computational complexity also.

2.4.3 Sample Spacing based Estimator

Sample-spacing based entropy estimates are mainly designed for 1-dimensional random variables. If we have real valued IID samples $\{x_1, x_2, \cdots, x_n\}$ of a random variable X with the order statistics $\{x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}\}$, then the *m-spacing* entropy estimate is given by [164]:

$$\hat{H}(X) = \frac{1}{n} \sum_{k=1}^{n} \log \left(\frac{n}{2m} (x_{(k+m)} - x_{(k-m)}) \right), \tag{2.20}$$

where $x_{(k)} = x_{(1)}$ if $k \leq 1$, and $x_{(k)} = x_{(n)}$ if $k \geq n$. The convergence and optimality of this estimator has been well studied in the past and has been found to be consistent for several different distributions [14, 32, 50]. One of the drawbacks of this technique is that the notion of order statistics is not well defined for a multi-dimensional random variable, therefore this method is mainly used in the 1-D case. However, recently some multi-dimensional *m-spacing* based entropy estimators have been developed that first map the high-dimensional random variables to Vornoi regions [84, 85].

2.4.4 Nearest Neighbour Estimator

The nearest neighbour estimator of entropy is defined for any multi-dimensional random variable X. It is based on the distance between the nearest neighbours of the realizations of random variables. If $\{x_1, x_2, \cdots, x_n\}$ are observations of a d dimensional random variable X, then the nearest neighbour estimate of entropy of X is given by [77]:

$$\hat{H}(X) = \frac{1}{n} \sum_{k=1}^{n} \log(n\rho_k) + \log 2 + C_E, \tag{2.21}$$

where ρ_i is the distance of x_i from its nearest neighbour x_j such that:

$$\rho_i = \min_{j \neq i, j \leq n} \|x_i - x_j\|, \tag{2.22}$$

and C_E is the Euler constant ($-\int_0^\infty \mathrm{e}^{-\mathrm{t}} \log \mathrm{t} \ \mathrm{dt}$). A *k*-nearest neighbour variant of the estimator was proposed by Singh et al. [151]. The statistical properties of the *k*-nearest

neighbour estimator has been extensively studied [44, 88, 154, 161], and it has been proved to be asymptotically unbiased and consistent.

2.4.5 Entropic Spanning Graphs

Another class of estimator uses the length of the Minimum Spanning Tree (MST) of the random samples to directly estimate the entropy. These methods are based on nearest neighbour graphs of the random samples, also called Entropic Spanning Graphs [57], and are *nonplug-in* estimators that do not estimate the probability distribution of the random variables. The Renyi entropy [135] of a d-dimensional random variable obtained from the MST graph is given by [58]:

$$\hat{H}_\alpha(X) = \frac{1}{1-\alpha}\big[\log \frac{L_\gamma(X)}{n^\alpha} - \log \beta\big], \tag{2.23}$$

where $L_\gamma(X)$ is the length of the MST, γ depends on the dimensionality d and order α of the Renyi entropy: $\gamma = d - \alpha d$, and β is the bias correction term and it depends on the graph minimization criteria used.

2.5 Conclusion

In this chapter we reviewed the concepts from probability and information theory that will be used extensively in subsequent chapters. We started with the basic concepts of probability theory and discussed how any real-world problem can be formulated in a probabilistic framework. We also discussed the uncertainty in any process and various measures of quantifying this uncertainty. Entropy is one of the most commonly used measures of uncertainty in a random variable and it forms the basis of information theory. Various information theoretic measures of statistical dependence of random variables (like MI) can be easily derived from entropy. We also provided a review of various estimators of entropy that have been well studied in the past and have been successfully used in practical applications. Information theoretic measures have been successfully used in communications, cryptography, bioinformatics, medical imaging and various other disciplines of science. The work in this thesis is mainly inspired by the applications of information theoretic concepts in medical imaging for registration of multi-modal data. In the field of medical imaging it is common to register images generated from different modalities like MRI, PET, etc. Fusion of these images allows experts to jointly analyze different information provided by these modalities. Registration of multi-modal data by optimizing certain information theoretic measures (MI, joint entropy, KL-divergence, etc) has become state-of-the-art in medical

imaging and is widely used in various clinical applications. However, these techniques have not received enough attention in the robotics community despite the fact that robots today are generally equipped with different sensing modalities. We believe that fusing the data obtained from these sensors can greatly enhance the robustness of robotics algorithm that require registration of multi-modal sensor-data. In the following chapters we will formulate some of the common robotics problems in an information theoretic framework and apply the concepts discussed in this chapter to solve those problems.

CHAPTER III

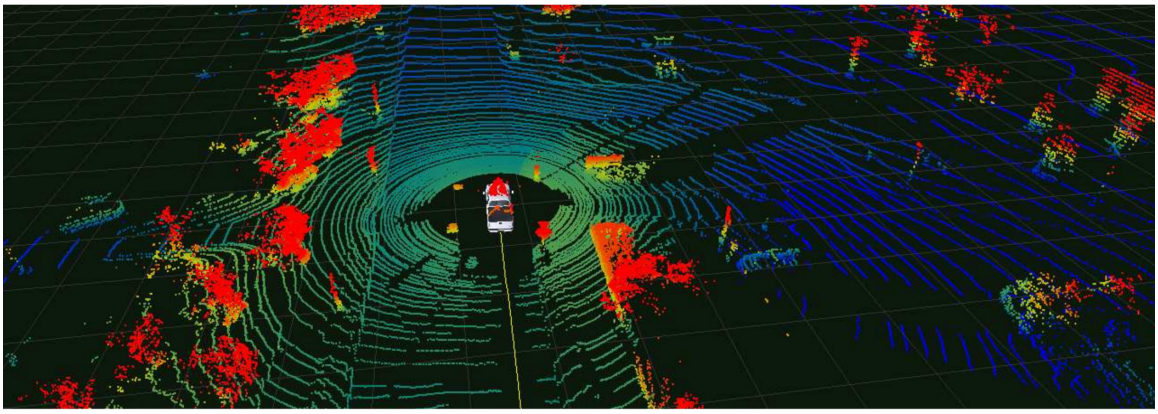
Extrinsic Calibration of Camera and Lidar

3.1 Introduction

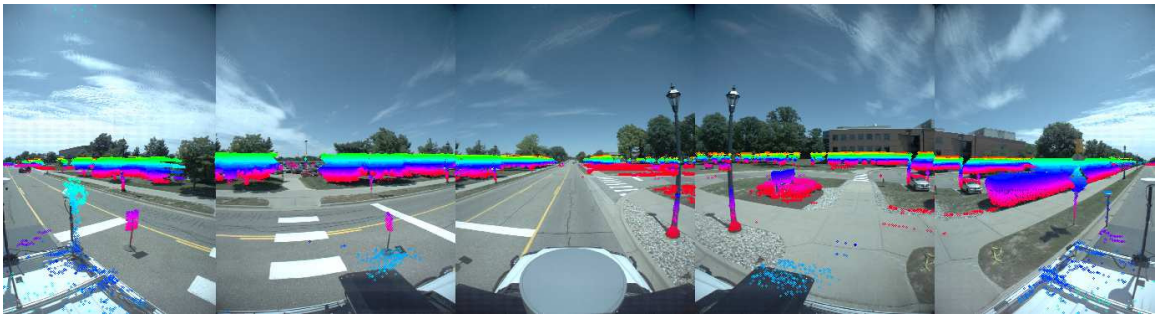
With recent advancements in sensing technologies, the ability to equip a robot with multi-sensor lidar/camera configurations has greatly improved. Two important categories of perception sensors commonly mounted on a robotic platform are: *(i)* range sensors (e.g., 3D/2D lidars, radars, sonars) and *(ii)* optical cameras (e.g., perspective, stereo, omnidirectional). Oftentimes the data obtained from these sensors is used independently; however, these modalities capture complementary information about the environment, which can be co-registered by extrinsically calibrating the sensors. This co-registration forms the basis for fusion of data obtained from the different modalities and is utilized in the subsequent chapters.

Extrinsic calibration is the process of estimating the rigid-body transformation between the reference coordinate system of the two sensors. This rigid-body transformation allows reprojection of the 3D points from the range sensor coordinate frame to the 2D camera coordinate frame (Fig. 3.1). Fusion of data provided by range and vision sensors can enhance various state-of-the-art computer vision and robotics algorithms. For example, Bao and Savarese [11] have proposed a novel framework for structure-from-motion (SFM) that takes advantage of both semantic (from camera data) and geometrical properties (from lidar data) associated with the objects in the scene. Nie et al. [119] proposed a road intersection detection method for path planning and control of an autonomous vehicle by fusing data from both lidar and camera modalities. Premebida et al. [133] proposed a pedestrian detection system for an autonomous vehicle in urban scenarios using information from lidar and a monocular camera. In mobile robotics, simultaneous localization and mapping (SLAM) is one of the basic tasks performed by robots. Although using a lidar for pose estimation and a camera for loop closure detection is common practice in SLAM [118], several successful attempts have been made to use the co-registered data in the SLAM framework

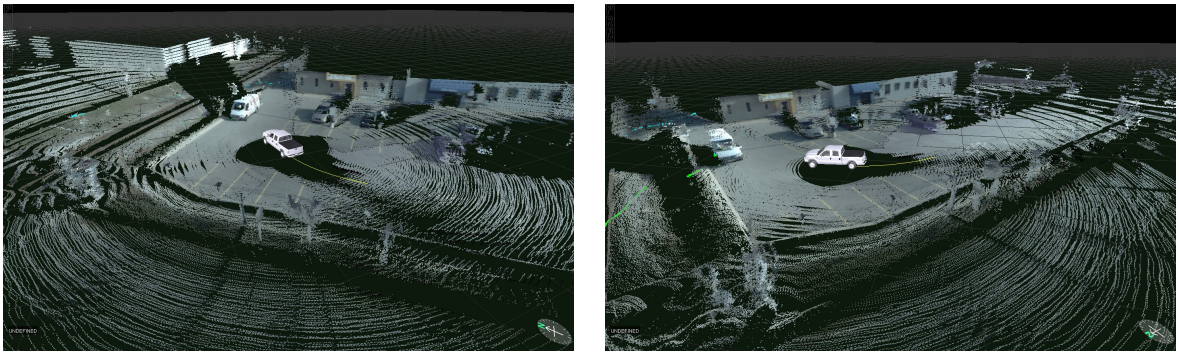
Figure 3.1 Reprojection of lidar and camera via extrinsic rigid-body calibration. (a) Perspective view of the 3D lidar range data, color-coded by height above the ground plane. (b) Depiction of the 3D lidar points projected onto the time-corresponding omnidirectional camera image. Several recognizable objects are present in the scene (e.g., people, stop signs, lamp posts, trees). Only nearby objects are projected for visual clarity. (c) Depiction of two different views of a fused lidar/camera textured point cloud. Each 3D point is colored by the RGB value of the pixel corresponding to the projection of the point onto the image.



(a) 3D lidar point cloud

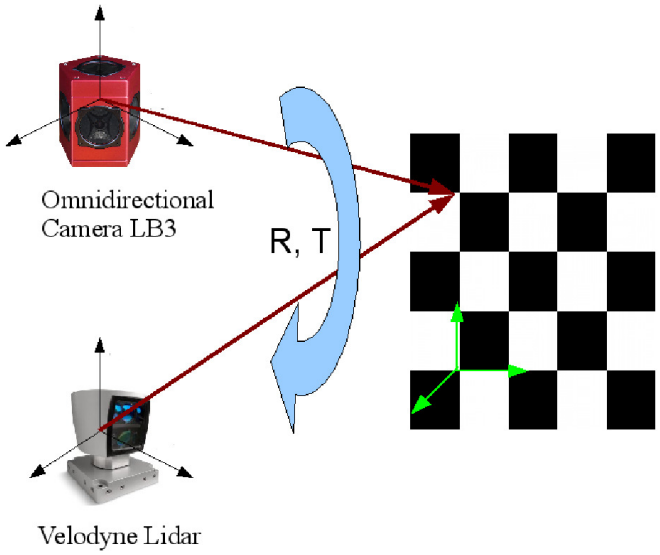


(b) Omnidirectional image with a subset of lidar points projected



(c) Fused RGB textured point cloud

Figure 3.2 Typical target-based calibration setup for an omnidirectional camera and a 3D lidar using a planar checkerboard pattern.



directly. Carlevaris-Bianco et al. [22] proposed a novel mapping and localization framework that uses the co-registered omnidirectional-camera imagery and lidar data to construct a map containing only the most viewpoint-robust visual features and then uses a monocular camera alone for online localization within the *a priori* map. Tamjidi and Ye [160] reported a six degree of freedom (DOF) vehicle pose estimation algorithm that uses fusion of lidar and camera data in both the feature initialization and motion prediction stages of an extended Kalman filter (EKF).

Extrinsic calibration is a core requisite for gathering useful data from a multi-sensor platform. Many of the existing algorithms for extrinsic calibration of lidar-camera systems require that fiducial targets be placed in the field of view of the two sensors. A planar checkerboard pattern (Fig. 3.2) is the most common calibration target used by researchers, as it is easy to extract from both camera and lidar data [173, 103, 163, 126]. The correspondences between lidar and camera data (e.g., point-to-point or point-to-plane) are established either manually or automatically and calibration parameters are estimated by minimizing a reprojection error. The accuracy of these methods is dependent upon the accuracy of the established correspondences. There are also methods that do not require any special targets but which rely upon extraction of some features (e.g., edges, lines, corners) from the camera and lidar data, either manually or automatically [142, 110, 91, 114]. The automatic feature extraction methods are generally not robust and require manual supervision to achieve small calibration errors. Although these methods can provide a good estimate of the calibration parameters, they are generally laborious and time consuming. Therefore,

due to the strenuous nature of the task, sensor calibration for a robotic platform is generally undertaken only once, assuming that the calibration parameters will not change over time. This may be a valid assumption for static platforms, but it is often not true for mobile platforms, especially in robotics. In mobile robotics, robots often need to operate in rough terrains, and assuming that the sensor calibration is not altered during a task is often not true.

In this chapter we discuss both target-based and target-less techniques of sensor calibration. In section 3.2 we describe a calibration technique that requires a checkerboard pattern viewed from the two sensors simultaneously. In section 3.3 we describe an algorithm for automatic, targetless, extrinsic calibration of a lidar and camera system that is suitable for easy in-field calibration. In section 3.4 we perform various experiments on real and simulated data and compare the results obtained from the two methods. In section 3.5 we present some concluding remarks.

3.2 Target-based Calibration

Several methods have been proposed in the past decade that use special calibration targets. One of the most common calibration targets used by researchers, a planar checkerboard pattern, was first used by Zhang [173] to calibrate a 2D laser scanner and a monocular camera system. He showed that the laser points lying on the checkerboard pattern and the normal of the calibration plane estimated in the camera reference frame provides a geometric constraint on the rigid-body transformation between camera and laser system. The transformation parameters are estimated by minimizing a nonlinear least squares cost function, formulated by reprojecting the laser points onto the camera image. This was probably the first published method that addressed the problem of extrinsic calibration of lidar/camera sensors in a robotics context. Thereafter, several modifications of Zhang's method have been proposed.

Mei and Rives [103] reported a similar algorithm for the calibration of a 2D laser range finder and an omnidirectional camera for both visible (i.e., laser is visible in camera image also) and invisible lasers. Zhang's method was later extended to calibrate a 3D laser scanner with a camera system [163, 126]. Nunnez et al. [121] modified Zhang's method to incorporate the inertial data from an inertial measurement unit (IMU) into the nonlinear cost function to increase the robustness of the calibration. Mirzaei et al. [108] provided an analytical solution to the least squares problem by formulating a geometric constraint between the laser points and the plane normal. This analytical solution was further improved by iteratively minimizing the nonlinear least squares cost function. The geometric

constraint in planar checkerboard methods requires the estimation of plane normals from camera and laser data. Therefore, the calibration error is correlated to the errors associated to the estimation of these plane normals.

In order to minimize this error, Zhou and Deng [175] proposed a new geometric constraint that decouples the estimation of rotation from translation by shifting the origin of the coordinate frame attached to the planar checkerboard target. Recently, Li et al. [93] proposed an algorithm for extrinsic calibration of a binocular stereo vision system and a 2D lidar. Instead of calibrating each camera of the stereo system independently with the lidar, they proposed an optimal extrinsic calibration method for the combined multi-sensor system based upon 3D-reconstruction of the checkerboard target. Although a planar checkerboard target is most common, several other specifically designed calibration targets have also been used in the past. Li et al. [92] designed a right-angled triangular checkerboard target and used the intersection points of the laser range finder's slice plane with the edges of the checkerboard to set up the constraint equation. Rodriguez et al. [137] used a circle-based calibration object to estimate the rigid-body transformation between a multi-layer lidar and camera system. Gong et al. [42] proposed an algorithm to calibrate a 3D lidar and camera system using geometric constraints associated with a trihedral object. Alempijevic et al. [3] reported a Mutual Information (MI)-based calibration framework that requires a moving object to be observed in both sensor modalities.

In the following section we describe an extrinsic calibration technique similar to the one proposed by Zhang [173], which requires the system to observe a planar pattern in several poses, and the constraints are based upon data captured simultaneously from the camera and the laser scanner. Zhang [173] presented results from a monocular camera system and a 2D laser scanner. Here we have extended Zhang's method to the case where we have a 3D laser scanner (Velodyne [165]) and an omnidirectional camera system (Ladybug3 [82]). The camera is pre-calibrated from the manufacturer so that the intrinsic parameters of individual camera are well known. Moreover, the rigid body transformation of all the cameras with respect to a common coordinate frame called the camera head are also known. We also discuss the possible degenerate cases and the minimum number of views of a planar checkerboard pattern required to be observed simultaneously from the laser scanner and the camera system. The normal of the planar surface and 3D points lying on the surface constrain the relative position and orientation of the laser scanner and the omnidirectional camera system. We show that these constraints can not only be used to form a non-linear optimization problem that is solved for the extrinsic calibration parameters but we can also use them to calculate the covariance associated with the estimated parameters.

3.2.1 Mathematical Formulation

The checkerboard pattern (target plane) is placed in the overlapping field-of-view of the two sensors (Fig. 3.2) so that it is observable in the data obtained from both sensors. The normal to the target plane and the laser points on the target plane are related, and constrain the relative position and orientation of the camera and laser scanner. We know the equation of the target plane in the coordinate system attached to the plane itself, which for convenience is given by:

$$Z = 0. \tag{3.1}$$

Let ${}^{w}\mathbf{P}$ be the coordinate of any point in the world reference frame (here it is the coordinate frame attached to the target plane) and ${}^{c_i}_{w}\mathrm{R}$ be the orthonormal rotation matrix that rotates frame w (world frame) into frame c_i (ith camera of the omni-directional camera system) and ${}^{c_i}\mathbf{t}_{c_i w}$ be the Euclidean 3-vector from c_i to w as expressed in frame c_i. Then the transformation equation that transforms a point from the world reference frame to the reference frame of the ith camera can be written as:

$${}^{c_i}\mathbf{P} \;=\; {}^{c_i}_{w}\mathrm{R}\,{}^{w}\mathbf{P} \;+\; {}^{c_i}\mathbf{t}_{c_i w}, \tag{3.2}$$

where ${}^{c_i}\mathbf{P}$ is the coordinate of that same point in ith camera's reference frame. Since we know the transformation matrices ${}^{h}_{c_i}\mathrm{R}$ and ${}^{h}\mathbf{t}_{hc_i}$ that transform a point from the ith camera frame to the camera head frame, we can write the coordinate of this point in the camera head frame as:

$${}^{h}\mathbf{P} \;=\; {}^{h}_{c_i}\mathrm{R}\,{}^{c_i}\mathbf{P} \;+\; {}^{h}\mathbf{t}_{hc_i}. \tag{3.3}$$

Thus, we can transform any point ${}^{w}\mathbf{P}$, lying in the target plane, into the camera head reference frame if we know the transformation ${}^{c_i}_{w}\mathrm{R}$ and ${}^{c_i}\mathbf{t}_{c_i w}$. We used Zhang's method [174] for finding this transformation relative to the planar target.

For a usual pin hole camera model, the relationship between a homogeneous 3D point ${}^{w}\tilde{\mathbf{P}} = [x\ y\ z\ 1]^{\top}$ and its image projection $\tilde{\mathbf{p}} = [u\ v\ 1]^{\top}$ is given by:

$$\tilde{\mathbf{p}} = \mathrm{K}_i[{}^{c_i}_{w}\mathrm{R}\ {}^{c_i}\mathbf{t}_{c_i w}]\,{}^{w}\tilde{\mathbf{P}}, \tag{3.4}$$

where $({}^{c_i}_{w}\mathrm{R}, {}^{c_i}\mathbf{t}_{c_i w})$, called the extrinsic parameters, are the rotation and translation that relates the world coordinate system to the camera coordinate system, and K_i is the camera intrinsic matrix.

Assuming that the image points are corrupted by independent and identically distributed noise, the maximum likelihood estimate of the required transformation $({}^{c_i}_{w}\mathrm{R}, {}^{c_i}\mathbf{t}_{c_i w})$ can be

obtained by minimizing the following reprojection error for n images of the target plane and m points per image [174]:

$$\underset{{}^{c_i}_{w}\mathrm{R},{}^{c_i}\mathbf{t}_{c_iw}}{\operatorname{argmin}} \sum_{k=1}^{n}\sum_{j=1}^{m} \|\tilde{\mathbf{p}}_{kj} - \mathrm{K}_i[{}^{c_i}_{w}\mathrm{R}{}^{c_i}\mathbf{t}_{c_iw}]{}^{w}\mathbf{P}_j\|. \tag{3.5}$$

Here, ${}^{c_i}_{w}\mathrm{R}$ is an orthonormal rotation matrix parametrized by the 3 Euler angles. Now, if ${}^{c_i}_{w}\mathrm{R} = [\mathbf{r_1}, \mathbf{r_2}, \mathbf{r_3}]$ and ${}^{c_i}\mathbf{t}_{c_iw}$ is the Euclidean 3-vector from c_i to w as expressed in frame c_i then we can write the equation of the target plane in the ith camera frame as:

$$\mathbf{r_3} \cdot (\mathbf{p} \; - \; {}^{c_i}\mathbf{t}_{c_iw}) = 0, \tag{3.6}$$

where $\mathbf{p}$ is the vector from the origin to any point lying on the plane. Therefore, the normal of the target plane in the ith camera frame is given by:

$${}^{c_i}\mathbf{N} = (\mathbf{r_3} \; \cdot \; {}^{c_i}\mathbf{t}_{c_iw})\mathbf{r_3}. \tag{3.7}$$

Here, $\|{}^{c_i}\mathbf{N}\| = \mathbf{r_3} \; \cdot {}^{c_i}\mathbf{t}_{c_iw}$ is the distance of the target plane from the ith camera's center. Since we know the pose of the ith camera with respect to the camera head we can calculate the normal of the plane ${}^{h}\mathbf{N}$ in the camera head frame as:

$${}^{h}\mathbf{N} \; = \; \frac{{}^{h}_{c_i}\mathrm{R}{}^{c_i}\mathbf{N}}{\|{}^{c_i}\mathbf{N}\|} \left(\|{}^{c_i}\mathbf{N}\| + \; \frac{{}^{c_i}\mathbf{N} \cdot \; {}^{h}\mathbf{t}_{hc_i}}{\|{}^{c_i}\mathbf{N}\|} \right). \tag{3.8}$$

Once we know the normal vector to the target plane in the camera head's reference frame, we need to find the 3D points in the laser reference frame that lie on the target plane. We use a RANSAC plane fitting algorithm to compute these 3D points. We also know the normal vector to the target plane from (3.8). These two measures provide a constraint on the required 3D rigid-body transformation between the laser and the camera system. Let $\{{}^{\ell}\mathbf{P}_i; i = 1, 2, \cdots, n\}$ be the set of 3D points lying on the plane given by RANSAC; the coordinates of these points are known in the laser reference system. The coordinates of these points in the camera head's frame are given by:

$${}^{h}\mathbf{P}_i \; = \; {}^{h}_{\ell}\mathrm{R}{}^{\ell}\mathbf{P}_i \; + \; {}^{h}\mathbf{t}_{h\ell}, \tag{3.9}$$

where ${}^{h}_{\ell}\mathrm{R}$ and ${}^{h}\mathbf{t}_{h\ell}$ are the required rotation and translation matrices that project any point in the laser reference system to the camera head's frame and thereby to the respective camera. Now, if we shoot a ray from the camera head to any point ${}^{h}\mathbf{P}_i$ lying on the plane, the projection of this ray onto the normal of the plane is equal to the distance of the plane from

the origin. Therefore, for m different views of the target plane and n 3D laser points per view, the laser-camera extrinsic parameters can be obtained by minimizing the following reprojection error:

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\frac{{}^{h}\mathbf{N}_i}{\|{}^{h}\mathbf{N}_i\|} \cdot ({}^{h}_{\ell}\mathrm{R} {}^{\ell}\mathbf{P}_j \ + \ {}^{h}\mathbf{t}_{h\ell}) - \|{}^{h}\mathbf{N}_i\| \right)^2, \tag{3.10}$$

where ${}^{h}\mathbf{N}_i$ is the normal to the i^{th} pose of the target plane in the camera head's frame. We can solve the nonlinear optimization problem given in (3.10) for ${}^{h}_{\ell}\mathrm{R}$ and ${}^{h}\mathbf{t}_{h\ell}$ using the Levenberg Marquadrt algorithm [89, 99].

3.2.2 Covariance of the Estimated Parameters

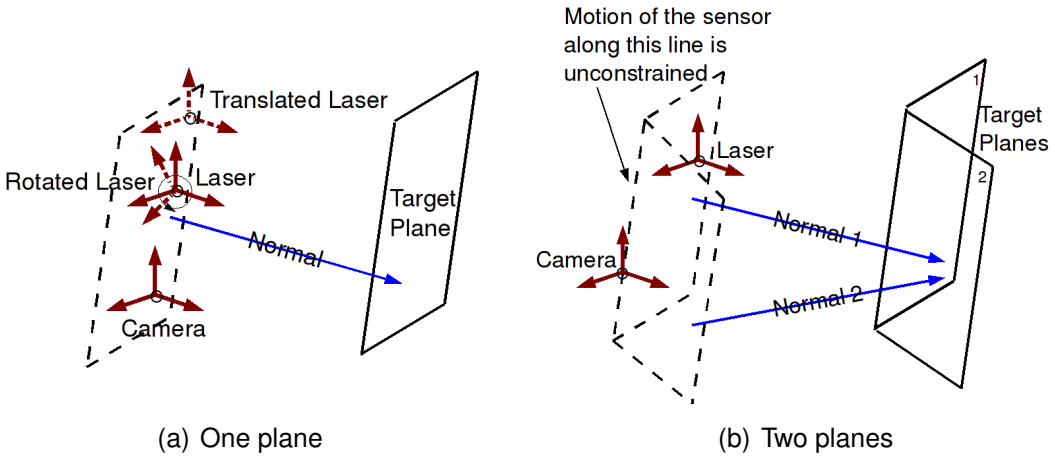
The parameters estimated by minimizing the cost function given in (3.10) have some error due to the uncertainty in the sensor measurements. The laser we have used in our experiments has uncertainty in the range measurements of the order of 0.02 m. This uncertainty due to the random perturbations of the range measurements is propagated to the estimated parameters. It is very important to know this uncertainty in order to use the parameters calculated here in any vision or SLAM algorithm. Haralick [52] has described a method to propagate the covariance of the measurements through any kind of scalar non-linear optimization function. The only assumptions are that the scalar function be non-negative, has finite first and second order partial derivatives, that its value be zero for ideal data, and the random perturbations in the input be small enough so that the output can be approximated by the first order Taylor series expansion. The optimization function (3.10) we use here satisfies these assumptions, so we can calculate the covariance of the estimated parameters as described by Haralick. Let us consider the laser-camera system such that the relative pose of the camera head with respect to the laser range finder be described by

$$\Theta = [{}^{\ell}\mathbf{t}_{\ell h}, \mathbf{\Phi}_{\ell h}]^{\top}. \tag{3.11}$$

Here, ${}^{\ell}\mathbf{t}_{\ell h} = [t_x, t_y, t_z]^{\top}$ is a Euclidean 3-vector from ℓ to h as expressed in frame ℓ, and $\mathbf{\Phi}_{\ell h} = [\theta_x, \theta_y, \theta_z]^{\top}$ is a 3-vector of zyx-convention roll, pitch, heading Euler angles that parametrizes the orthonormal rotation matrix ${}^{\ell}_{h}\mathrm{R}$ (which rotates frame h into frame ℓ). The covariance of the estimated parameters Θ can thus be given as:

$$\Sigma_{\Theta} = \left[\frac{\partial^2 F}{\partial \Theta^2}(X, \Theta) \right]^{-1} \frac{\partial^2 F^{\top}}{\partial X \partial \Theta}(X, \Theta) \Sigma_X \frac{\partial^2 F}{\partial X \partial \Theta}(X, \Theta) \left[\frac{\partial^2 F}{\partial \Theta^2}(X, \Theta) \right]^{-1} \tag{3.12}$$

Figure 3.3 Geometrical interpretation of minimum number of views required for calibration. (a) The translation of the sensors along the target plane and rotation about the axis parallel to normal of the plane is not constrained. (b) The translation of the sensors along the line of intersection of the two planes is not constrained.



Here, $X = [{}^{h}\mathbf{N}_1, \{{}^{\ell}\mathbf{P}\}_1, \ldots {}^{h}\mathbf{N}_i, \{{}^{\ell}\mathbf{P}_1\}, \ldots]^{\top}$ is the vector of measurements composed of the normals of the planes observed (${}^{h}\mathbf{N}_i$) and the laser points lying on these planes ($\{{}^{\ell}\mathbf{P}\}_i = \{{}^{\ell}p_1, {}^{\ell}p_2, \cdots\}_i$). Please see Appendix B for implementation details.

3.2.3 Minimum Number of Views Required

A minimum of three non-coplanar views of the target plane are required to fully constrain the optimization problem (3.10) for the estimation of the calibration parameters. If only one plane is considered, as shown in Fig. 3.3(a), then the cost function (3.10) does not change when the sensors are either translated along the plane parallel to the target plane or rotated about the axis parallel to the normal of the target plane. Thus, the solution obtained from a single view does not converge to the actual value in the following three parameters: 2D translation along the target plane and a rotation about the normal of the target plane. Similarly for two views (Fig. 3.3(b)) the translation of the sensor along the line of intersection of the two planes does not change the cost function, thereby giving large uncertainty in that direction. Three views are required to completely constrain the 6 degree of freedom (DOF) pose of one sensor with respect to the other.

Although we show that only three views are sufficient to estimate the rigid-body transformation between the laser-camera system, in practice we need significantly large number of views. In Fig. 3.4 we show that the estimation error decreases with the increase in the number of views of the target plane. Increasing the number of views increases the number of constraints in the optimization (3.10). Since the omnidirectional camera system is com-

posed of six different cameras, we should take a sufficient number of planes viewed from all the cameras so that our estimate is not biased toward any one camera of the system. Moreover, the size or area of the target plane also plays a significant role in ensuring the accuracy of the estimated parameters. As shown in Fig. 3.5 the estimation error decreases as the area of the planar surface increases. This is because when we have a larger surface area the number of 3D laser points falling on the plane increases thereby increasing the number of constraints in the optimization (3.10). In practice we can stick a small ($1\ \mathrm{m} \times 1\ \mathrm{m}$) checkerboard pattern on the walls available in the experimental site to get large target planes for the 3D laser data.

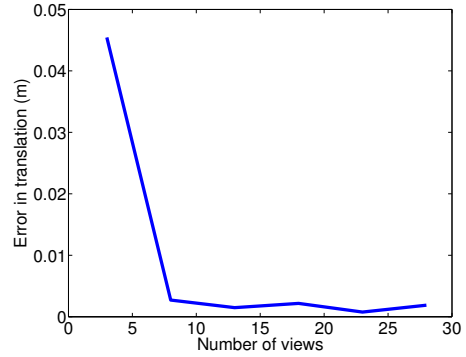
3.3 Target-less Calibration

The target-based methods require a fiducial object to be concurrently viewed from the lidar and camera sensors, and are in general very laborious and time-consuming. Therefore they are not practical for easy *in-situ* calibration. This is the reason why sensor calibration, when performed with the target based technique, in a robotic application is typically performed once, and the same calibration is assumed to be true for rest of the life of that particular sensor suite. However, for robotics applications where the robot needs to go out into rough terrain, assuming that the sensor calibration is not altered during a task is often not true. The errors introduced due to the assumption that *the calibration does not change* can easily break any robotic system which depends upon the sensor calibration. Therefore, we need automatic methods of sensor calibration that can be used to fine tune the calibration of the sensors *in situ*.

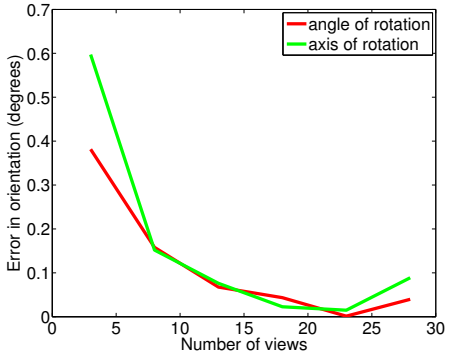
Scaramuzza et al. [142] introduced a target-less technique for the calibration of a 3D laser scanner and omnidirectional camera from natural scenes. They automatically extracted some features from the camera and lidar data and then manually established correspondence between the extracted features. The calibration parameters were then estimated by minimizing the reprojection error for the corresponding points. Recently, Moghadam et al. [110] proposed a method that exploits the linear features present in a typical indoor environment. The 3D line features extracted from the point cloud and the corresponding 2D line segments extracted from the camera images are used to constrain the rigid-body transformation between the two sensor coordinate frames.

There are also techniques that exploit the statistical dependence of the data measured from the two sensors to obtain a calibration. Boughorbal et al. [19] proposed a χ^2 test that maximizes the correlation between the sensor data to estimate the calibration parameters. A similar technique was later used by Williams et al. [168], but their method requires addi-

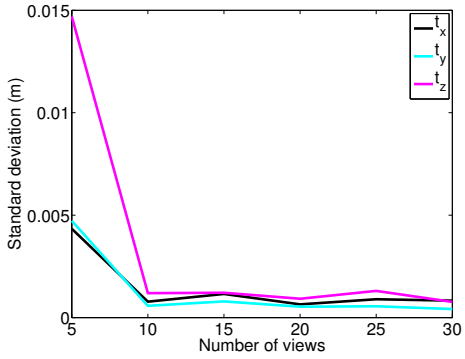
Figure 3.4 We Simulated certain number of views of a planar surface, randomly generated around a unit sphere of the laser-camera system. The laser points lying on the simulated plane are computed based on the relative pose of the laser and camera head. We then add uniform Gaussian noise of 10 cm to the range measurements of the laser points. These noisy points are then used to estimate the calibration parameters. The plotted error in estimation of calibration parameters decreases as the number of views increases. Area of the simulated plane is fixed to 1 m^2.



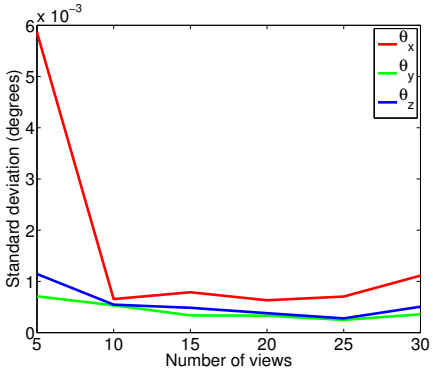
(a) Error in translation



(b) Error in rotation



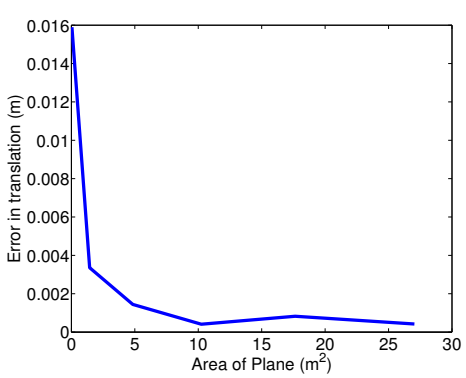
(c) Standard deviation of estimated translation parameters



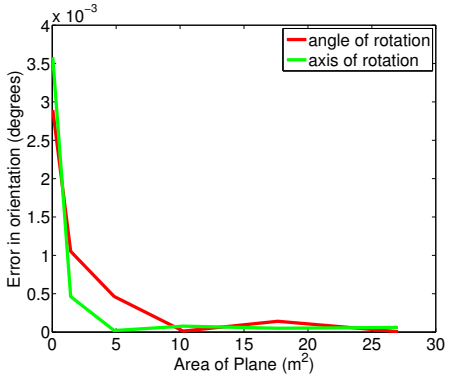
(d) Standard deviation of estimated rotation parameters

tional techniques to estimate the initial guess of the calibration parameters. Levinson and Thrun [91] use a series of corresponding laser scans and camera images of arbitrary scenes to automatically estimate the calibration parameters. They use the correlation between the depth discontinuities in laser data and the edges in camera images. A cost function is formulated that captures the strength of the co-observation of depth discontinuity in laser data and corresponding edge in the camera image. Recently, Napier et al. [114] presented a method that calibrates a 2D push broom lidar and a camera system by optimizing a correlation measure between the laser reflectivity and grayscale values from the camera imagery

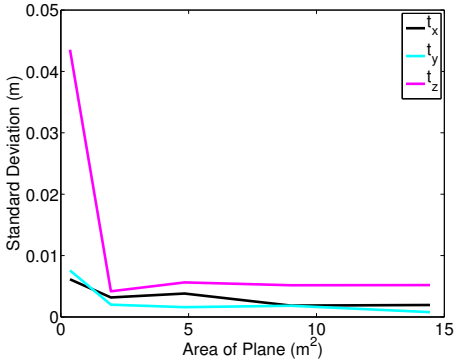
Figure 3.5 We Simulated fixed number of views of a planar surface, randomly generated around a unit sphere of the laser-camera system. The laser points lying on the simulated plane are computed based on the relative pose of the laser and camera head. We then add uniform Gaussian noise of 10 cm to the range measurements of the laser points. These noisy points are then used to estimate the calibration parameters. The plotted error in estimation of calibration parameters decreases as the area of target plane increases. We use a fixed number of views (10) for each trial.



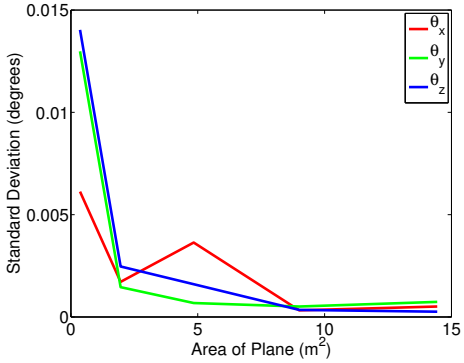
(a) Error in translation



(b) Error in rotation



(c) Standard Deviation of estimated translation parameters



(d) Standard Deviation of estimated rotation parameters

acquired from natural scenes. They do not require the sensors to be mounted such that they have overlapping field of view and compensate for it by observing the same scene at different times from a moving platform. Therefore, they require accurate measurements from an IMU mounted on the moving platform.

In the following section of this chapter we describe an automatic targetless algorithm for extrinsic calibration. The recent work by Levinson and Thrun [91] and Napier et al. [114] are closely related to the one presented here, in the sense that they also propose a fully automatic and targetless method for extrinsic calibration; however, their formulation of the

Figure 3.6 Various range sensors used in robotics applications

(a) 2D laser scanner (Hokuyo) (b) 3D laser scanner (Velodyne HDL64e) (c) TOF 3D camera (PMD CamCube) (d) 3D depth camera (Microsoft Kinect)

optimization function is quite different. As far as the method is concerned, Boughorbal et al. [19] and Williams et al. [168] are the most closely related previous works to the one described here, though they have reported problems of existence of local maxima in the objective-function formulated using either MI or χ^2. We have solved this problem by incorporating scans from different scenes into a single optimization framework, thereby obtaining a smooth and concave objective function that is easy to solve by any gradient ascent algorithm. We also show the robustness of the algorithm by performing several different experimental setups using real data obtained from a variety of range/image sensor pairs.

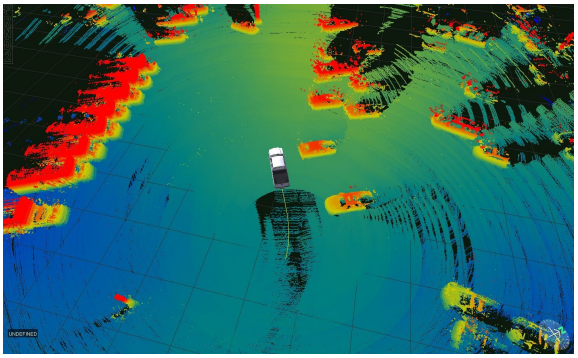
The proposed algorithm is completely data driven and can be used with any camera, and any range sensor that reports meaningful surface reflectivity values and scene depth information. Various range sensors commonly used in robotics and mapping applications are shown in Fig. 3.6. Most of these sensors report meaningful surface reflectivity values that can be directly used in the proposed algorithm, but for multi-beam sensors like the Velodyne [165], it is important to first perform inter-beam calibration of the surface reflectivity values [90]. Here, we assume that the reflectivity values are cross-beam calibrated wherever necessary.

We use the surface reflectivity values reported by the range sensor and the grayscale intensity values reported by the camera to extrinsically calibrate the two sensor modalities. We claim that under the correct rigid-body transformation, the correlation between the laser reflectivity and camera intensity is maximized. Our claim is illustrated by a simple experiment shown in Fig. 3.7. Here, we calculate the correlation coefficient for the reflectivity and intensity values for a scan-image pair at different values of the calibration parameter and observe a distinct maxima at the true value. Moreover, we observe that the joint histogram of the laser reflectivity and the camera intensity values is least dispersed when calculated under the correct rigid-body transformation.

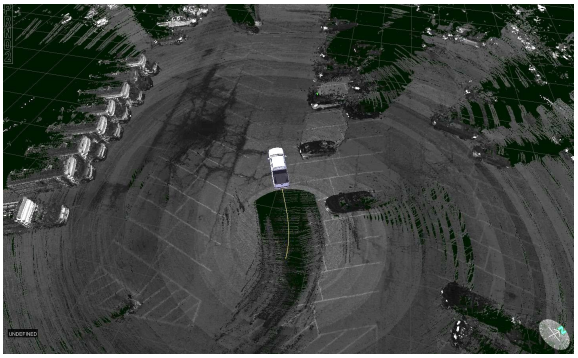
Although scenarios such as Fig. 3.7 do exhibit high correlation between the two modalities, there also exist counterexamples where the two modalities may not be as strongly

Figure 3.7 Simple experiment illustrating the available correlation between lidar measured surface reflectivity and camera measured image intensity. (a) Image from the Ladybug3 omnidirectional camera. (b) & (c) Depiction of the Velodyne HDL-64E 3D lidar data color-coded by height above ground and by laser reflectivity, respectively. (d) The correlation coefficient for the reflectivity/intensity values as a function of one of the extrinsic calibration parameters, pitch, while keeping all other parameters fixed at their true value. We observe that the correlation coefficient is maximum for the true pitch angle of $0°$, denoted by the dashed vertical line. (e) Depiction of the joint histogram of the reflectivity and intensity values when calculated at an incorrect (left) and correct (right) rigid-body transformation. Note that the joint histogram is least dispersed under the correct rigid-body transformation.

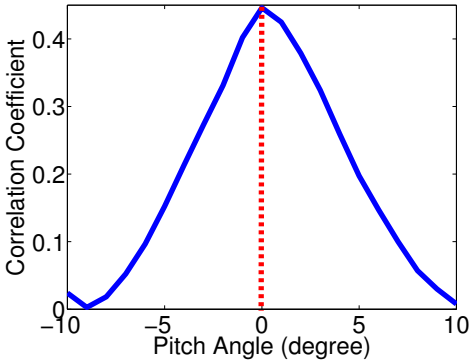
(a) Omnidirectional camera image



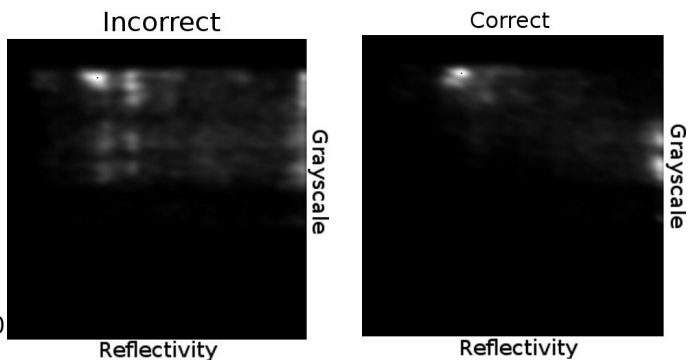
(b) Corresponding lidar colored by height



(c) Corresponding lidar colored by reflectivity



(d) Grayscale/reflectivity correlation



(e) Grayscale/reflectivity joint-distribution

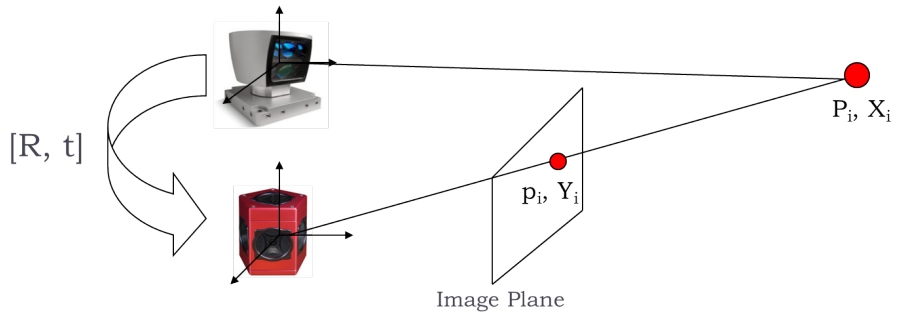
Figure 3.8 Counterexample showing that non-uniform lighting can play a critical role in influencing reflectivity/intensity correlation. (left) Ambient lit image with shadows of trees and buildings on the road. (right) Top view of the corresponding lidar reflectivity map, which is unaffected by ambient lighting due to its active lighting principle.

correlated, for example see Fig. 3.8. Here, ambient light plays a critical role in determining the intensity levels of image pixels on the road. As clearly depicted in the image, there are some regions of the road that are covered by object shadows. The gray levels of the image are locally affected by the shadows of occluding objects; however, the corresponding reflectivity values in the laser modality are not because it uses an active lighting principle. Thus, in these type of scenarios the data between the two sensors might not show as strong of a correlation and, hence, will produce a weak input for the proposed algorithm. Here we do not focus on solving the general lighting problem, instead, we formulate a MI-based data fusion criterion to estimate the extrinsic calibration parameters between the two sensors assuming that the data is, for the most part, not corrupted by lighting artifacts. In fact, for many practical indoor/outdoor calibration scenes (e.g., Fig. 3.7), shadow effects represent a small fraction of the overall data and thus appear as noise in the calibration process. This is easily handled by the proposed method by aggregating multiple scan views.

3.3.1 Mathematical Formulation

Here we consider the laser reflectivity value of a 3D point and the corresponding grayscale value of the image pixel to which this 3D point is projected as the random variables X and Y, respectively. The marginal and joint probabilities of these random variables, $p(X)$, $p(Y)$ and $p(X,Y)$, can be obtained from the normalized marginal and joint histograms of the reflectivity and grayscale intensity values of the 3D points co-observed by the lidar and camera. Let $\{\mathbf{P}_i;\, i = 1, 2, \cdots, n\}$ be the set of 3D points whose coordinates are known in the laser reference system and let $\{X_i;\, i = 1, 2, \cdots, n\}$ be the corresponding reflectivity values for these points ($X_i \in [0, 255]$).

Figure 3.9 Illustration of mathematical formulation of MI-based calibration.



For the usual pinhole camera model, the relationship between a homogeneous 3D point, $\tilde{\mathbf{P}}_i$, and its homogeneous image projection, $\tilde{\mathbf{p}}_i$, is given by:

$$\tilde{\mathbf{p}}_i = \mathrm{K}\big[\mathrm{R} \,|\, \mathbf{t}\big]\tilde{\mathbf{P}}_i, \tag{3.13}$$

where $(\mathrm{R}, \mathbf{t})$, called the extrinsic parameters, are the orthonormal rotation matrix and translation vector that relate the laser coordinate system to the camera coordinate system, and K is the camera intrinsics matrix. Here, R is parametrized by the Euler angles $[\phi, \theta, \psi]^\top$ and $\mathbf{t} = [x, y, z]^\top$ is the translation vector. Let $\{Y_i;\, i = 1, 2, \cdots, n\}$ be the grayscale intensity value of the image pixel upon which the 3D laser point projects such that

$$Y_i = \mathrm{I}(\mathbf{p}_i), \tag{3.14}$$

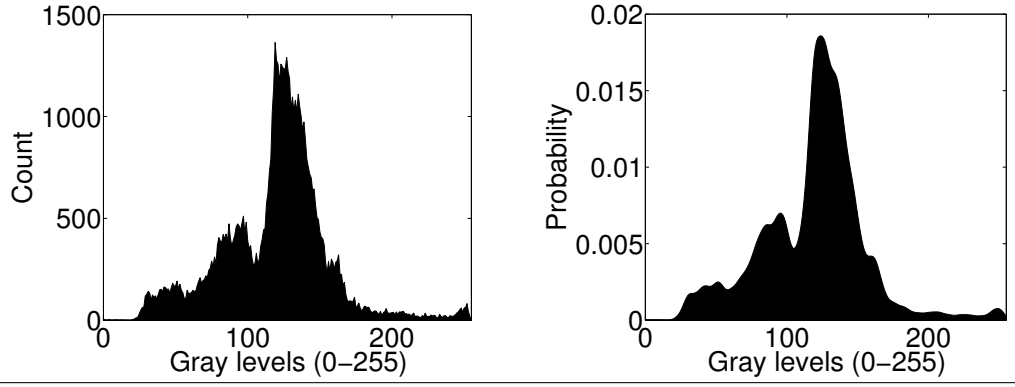
where $Y_i \in [0, 255]$, I is the grayscale image, and $\mathbf{p}_i$ is the inhomogeneous version of $\tilde{\mathbf{p}}_i$.

Thus, for a given set of extrinsic calibration parameters, X_i and Y_i are the observations of the random variables X and Y, respectively (Fig. 3.9). The marginal and joint probabilities of the random variables X and Y can be obtained from the kernel density estimate (KDE) of the normalized marginal and joint histograms of X_i and Y_i. The KDE of the joint distribution of the random variables X and Y is given by [145]:

$$p(X, Y) = \frac{1}{n}\sum_{i=1}^{n} K_\Omega\left(\begin{bmatrix} X \\ Y \end{bmatrix} - \begin{bmatrix} X_i \\ Y_i \end{bmatrix}\right), \tag{3.15}$$

where $K_\Omega(\,\cdot\,)$ is a symmetric kernel and Ω is the *bandwidth* or the *smoothing* matrix of the kernel. In our experiments we have used a Gaussian kernel and the bandwidth matrix

Figure 3.10 Kernel density estimate of the probability distribution (right), estimated from the observed histogram (left) of grayscale intensity values.



Ω is computed from the *Silverman's rule of thumb* [150]:

$$\Omega = 1.06n^{1/5} \begin{bmatrix} \sigma_X & 0 \\ 0 & \sigma_Y \end{bmatrix}, \tag{3.16}$$

where σ_X and σ_Y are the standard deviations of the observations of X and Y, respectively. An illustration of the KDE of the probability distribution of the grayscale values from the available histograms is shown in Fig. 3.10.

Once we have an estimate of the probability distribution we can write the MI of the two random variables as a function of the extrinsic calibration parameters $(\mathrm{R}, \mathbf{t})$, thereby formulating an objective function:

$$\hat{\mathbf{\Theta}} = \arg\max_{\mathbf{\Theta}} \ \mathrm{MI}(X, Y; \mathbf{\Theta}), \tag{3.17}$$

whose maxima occurs at the sought after calibration parameters, $\mathbf{\Theta} = [x, y, z, \phi, \theta, \psi]^\top$. The complete MI-based calibration algorithm is shown in Algorithm 1.

3.3.2 Optimization

The cost function (3.17) is maximized at the correct value of the rigid body transformation parameters. Therefore, any optimization technique that iteratively converges to the global optimum can be used here. Some of the commonly used optimization techniques compute the gradient or hessian of the cost function [167, 89, 99, 12]. Since, the proposed cost function does not have a parametric form, we can use numerical methods to compute the gradients. Some techniques do not even require the computation of gradients, but use heuristics to converge to the global optimum [115, 73, 37]. Moreover, one can even use exhaustive search to obtain the global optima of the cost function.

Algorithm 1 Automatic extrinsic calibration by maximization of Mutual Information

Input: 3D Point cloud $\{\mathbf{P}_i;\, i = 1, \cdots, n\}$, Reflectivity $\{X_i;\, i = 1, \cdots, n\}$, Image $\{\mathrm{I}\}$ and Initial guess $\{\mathbf{\Theta}_0\}$
Output: Estimated parameter $\{\hat{\mathbf{\Theta}}\}$
while $\|\mathbf{\Theta}_{k+1} - \mathbf{\Theta}_k\| > THRESHOLD$ **do**
  $\mathbf{\Theta}_k \rightarrow \big[\mathrm{R} \,|\, \mathbf{t}\big]$
  Initialize the joint histogram: $\mathrm{Hist}(X, Y) = 0$
  for $i = 1 \rightarrow n$ **do**
    $\tilde{\mathbf{p}}_i = \mathrm{K}\big[\mathrm{R} \,|\, \mathbf{t}\big]\tilde{\mathbf{P}}_i$
    $Y_i = \mathrm{I}(\mathbf{p}_i)$
    Update the joint histogram: $\mathrm{Hist}(X_i, Y_i) = \mathrm{Hist}(X_i, Y_i) + 1$
  end for
  Calculate the kernel density estimate of the joint distribution: $p(X, Y; \mathbf{\Theta}_k)$
  Calculate the Mutual Information: $\mathrm{MI}(X, Y; \mathbf{\Theta}_k)$
  Update the current estimate: $\mathbf{\Theta}_{k+1} = \Theta_k + \lambda F(\mathrm{MI}(X, Y; \mathbf{\Theta}_k))$, where F is either the gradient function or some heuristic which depends upon the choice of optimization technique and λ is a tuning parameter specific to that optimization algorithm.
end while

3.3.3 Cramér-Rao Lower Bound of the Estimated Parameter Variance

It is important to know the uncertainty in the estimated calibration parameters in order to use them in any vision or SLAM algorithm. Here we use the Cramer-Rao-Lower-Bound (CRLB) of the variance of the estimated parameters as a measure of the uncertainty. The CRLB [30] states that the variance of any unbiased estimator is greater than or equal to the inverse of the Fisher information matrix. Moreover, any unbiased estimator that achieves this lower bound is said to be efficient. The Fisher information of a random variable Z is a measure of the amount of information that the observations of the random variable Z carries about an unknown parameter α, upon which the probability distribution of Z depends. If the distribution of a random variable Z is given by $p(Z; \alpha)$, then the Fisher information is given by [86]:

$$\mathcal{I}(\alpha) = \mathrm{E}\left[\left(\frac{\partial}{\partial \alpha} \log p(Z; \alpha)\right)^2\right]. \tag{3.18}$$

In our case the joint distribution of the random variables X and Y, as defined in (3.15), depends upon the six dimensional transformation parameter $\mathbf{\Theta}$. Therefore, the Fisher information is given by a $[6 \times 6]$ matrix, $\mathcal{I}(\mathbf{\Theta})$, whose elements are individually computed as

$$\mathcal{I}(\mathbf{\Theta})_{ij} = \mathrm{E}\left[\frac{\partial}{\partial \mathbf{\Theta}_i} \log p(X, Y; \mathbf{\Theta}) \frac{\partial}{\partial \mathbf{\Theta}_j} \log p(X, Y; \mathbf{\Theta})\right]. \tag{3.19}$$

Figure 3.11 The modified Ford F-250 pickup truck with sensor configuration as described in appendix C. The 3D laser scanner [165] and the omnidirectional camera [82] are mounted on the roof of the vehicle.

The CRLB is then given by

$$\mathrm{Cov}(\mathbf{\Theta}) \geq \mathcal{I}(\mathbf{\Theta})^{-1}, \tag{3.20}$$

where $\mathcal{I}(\mathbf{\Theta})^{-1}$ is the inverse of the Fisher information matrix calculated at the estimated value of the parameter $\hat{\mathbf{\Theta}}$.

3.4 Experiments and Results

This section describes in detail various experiments performed to evaluate the accuracy and robustness of the proposed calibration techniques. We present both qualitative and quantitative results with data collected from three different sensor pairs commonly used in robotics applications.

3.4.1 3D Laser Scanner and Omnidirectional Camera

In the first set of experiments we present results from data collected from a 3D laser scanner [165] and an omnidirectional camera system [82] mounted on the roof of a vehicle (Fig. 3.11). In this work we pre-calibrated the reflectivity values of the Velodyne laser scanner using the algorithm reported by [90], and used the manufacturer provided intrinsic calibration parameters (focal length, camera center, distortion coefficients of the lens) for the omnidirectional camera. In all of our experiments in this section $scan$ refers to a single 360° field of view 3D point cloud and its time-corresponding camera imagery.

Figure 3.12 Setup for target-based calibration inside a garage. The left panel shows the checkerboard pattern mounted on a planar surface as seen from one of the cameras of the omnidirectional camera system. The estimated normal of the planar surface in that camera's reference system is shown in green. The right panel shows the 3D points lying on the planar checkerboard pattern in laser reference system.

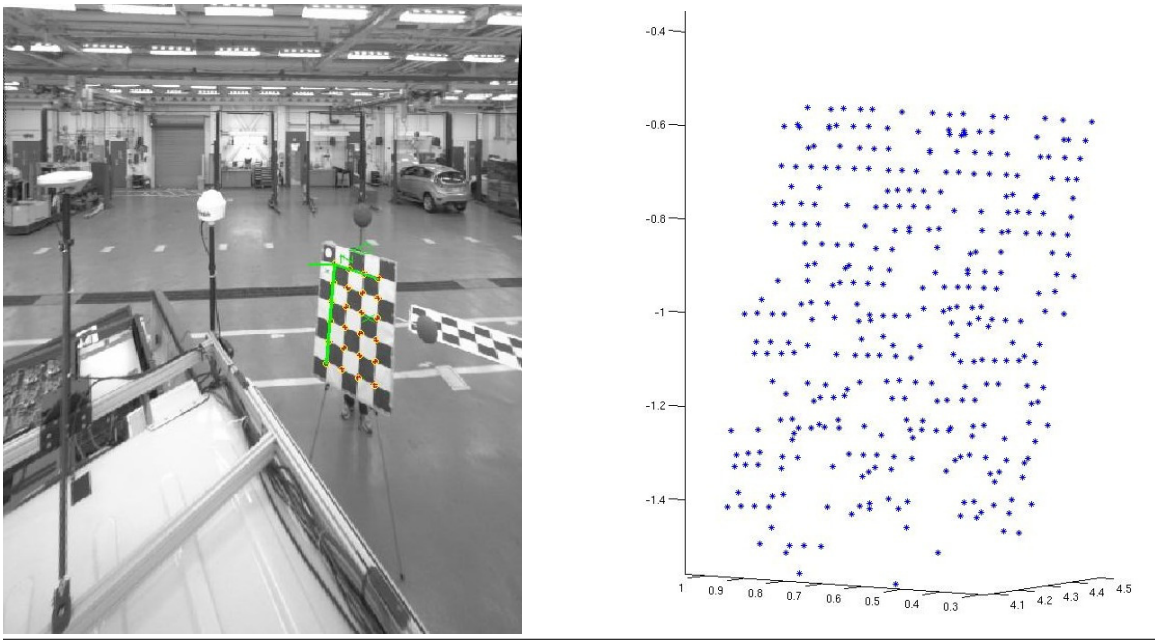
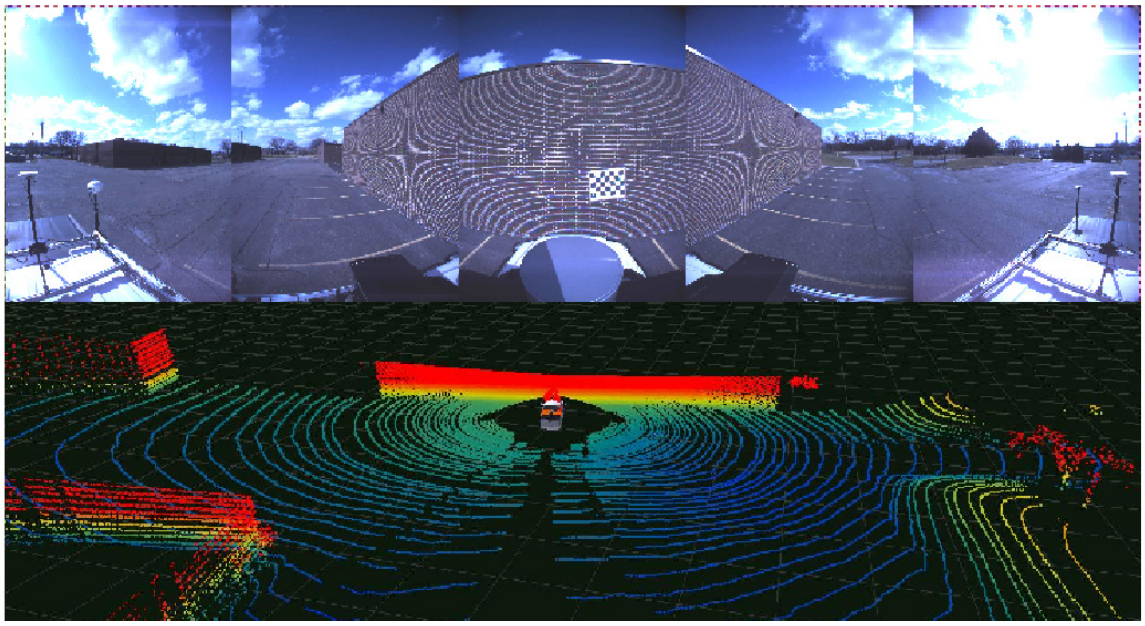


Figure 3.13 Vehicle parked in-front of a wall and the checkerboard pattern pasted on it. The top panel shows the image from the omnidirectional camera and the bottom panel shows the corresponding 3D point cloud from the lidar. The wall is used as a calibration target.



3.4.1.1 Target-based Calibration

In this experiment we calibrated the sensors by using planar checkerboard targets as described in section 3.2. The calibration data was collected inside a garage where checkerboard patterns were mounted on all available planar surfaces including side walls and ground floor (Fig. 3.12). We also used a planar checkerboard target of size $75\ \mathrm{cm} \times 105\ \mathrm{cm}$ and manually moved it around the field of view (FOV) of the two sensors in all possible orientations. It took us about 30 minutes to collect this dataset with one person moving around the vehicle holding the big checkerboard target (Fig. 3.12). The simulation results in section 3.2.3 showed that the calibration performance increases as we increase the area of the target plane. Therefore, we took the vehicle outside and parked it in-front of a huge wall; we pasted a checkerboard pattern on the wall and then moved the vehicle, so that the entire wall can be used as the calibration target (Fig. 3.13). The target-based method is in general very time consuming and involves significant manual intervention. The result obtained by the target-based method is shown in Table 3.1.

3.4.1.2 Targetless Calibration: Performance Using a Single Scan

In this experiment we show that the quality of the *in situ* calibration performance is dependent upon the environment in which the scans are collected. We collected several datasets in both indoor and outdoor settings. The indoor dataset was collected inside a large garage, and exhibited many near-field objects such as walls and other vehicles. In contrast, most of the objects in the outdoor dataset were far-field. In the presence of only far-field 3D points, the cost-function is insensitive to the translational calibration parameters—making them more difficult to estimate. This is a well-known phenomenon of projective geometry, where in the limiting case if we consider points at infinity, $[\tilde{x}, \tilde{y}, \tilde{z}, 0]^\top$, the projection of these points (also known as vanishing points) are not affected by the translational component of the camera projection matrix [53]:

$$\tilde{\mathbf{p}} = \mathrm{K}\big[\mathrm{R} \,|\, \mathbf{t}\big] \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 0 \end{bmatrix} = \mathrm{KR} \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix}. \tag{3.21}$$

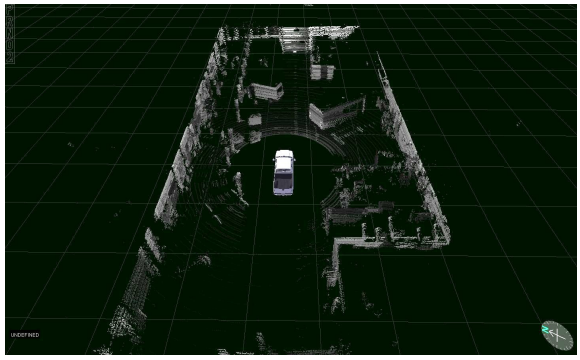
We should then expect that scans which only contain 3D points far-off in the distance (e.g., the outdoor dataset) will have poor observability of the extrinsic translation vector, $\mathbf{t}$, as opposed to scans that contain many nearby 3D points (e.g., the indoor dataset), as seen in Fig. 3.14. In Fig. 3.14(e) and (f) we have plotted the calibration results for 15 scans

Figure 3.14 3D laser and omnidirectional camera single-view calibration results for outdoor and indoor datasets. The variance in the estimated parameters (especially translation) is significantly large in the case of the outdoor dataset due to poor observability as noted in the text. Each point on the abscissa in (e)–(f) corresponds to a single scan trial.

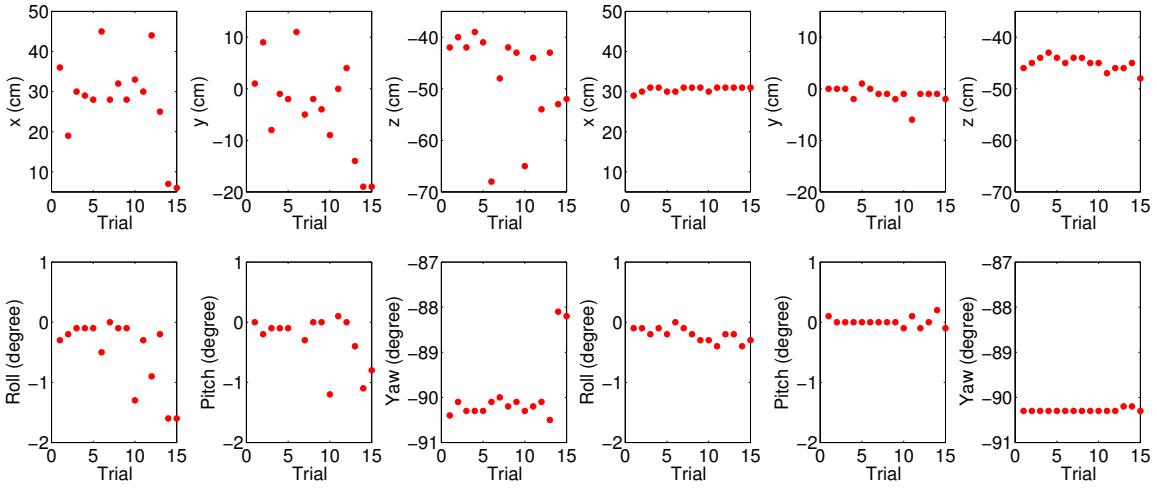
(a) Sample omnidirectional image (Outdoor)

(b) Sample omnidirectional image (Indoor)

(c) Sample laser scan (Outdoor)



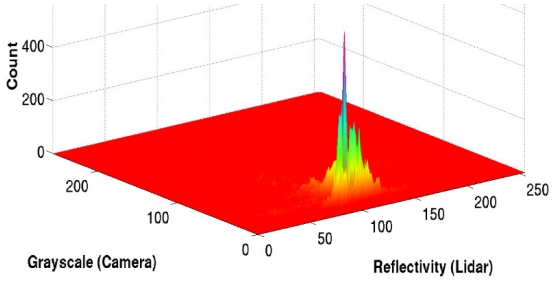
(d) Sample laser scan (Indoor)



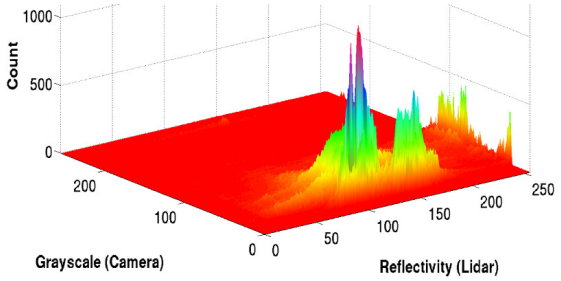
(e) MI based calibration results (Outdoor)

(f) MI based calibration results (Indoor)

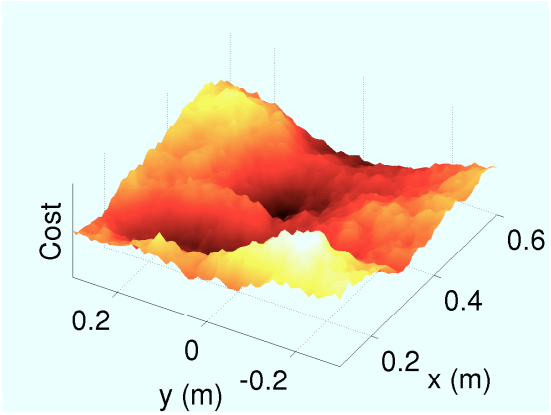
Figure 3.15 The top panel shows the joint-histogram of lidar reflectivity and camera intensity values. We get a better estimate of the joint histogram (fill-in of unobserved sections) as the number of scan-image pairs is increased. The bottom panel shows the MI cost-function surface versus translation parameters x and y. Note the global convexity and smoothness when the scans are aggregated. The correct value of parameters is given by (0.3, 0.0). Negative MI is plotted here to make visualization of the extrema easier.



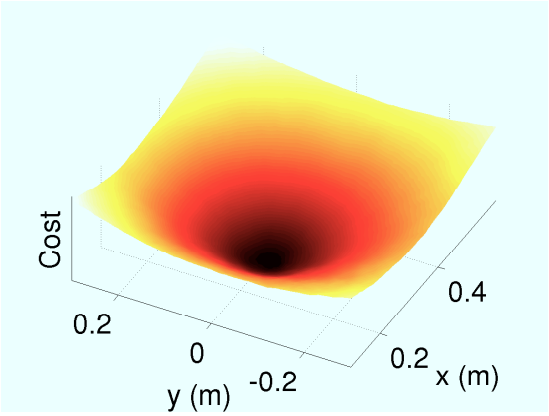
(a) Histogram with 1 scan



(b) Histogram with 10 scans



(c) Cost function with 1 scan



(d) Cost function with 10 scans

collected in outdoor and indoor settings, respectively. We clearly see that the variability in the estimated parameters for the outdoor scans is much larger than that of the indoor scans. Thus, from this experiment we conclude that we need to have near-field objects in order to robustly estimate the calibration parameters from a single-view.

3.4.1.3 Targetless Calibration: Performance Using Multiple Scans

In the previous section we showed that it is necessary to have near-field objects in the scans in order to robustly estimate the calibration parameters from a single-scan; however, this might not always be practical—depending upon the operational environment. In this experiment we demonstrate improved calibration convergence by simply aggregating multiple scans into a single batch optimization process (Fig. 3.15). It should be noted that the reflectivity from lidar and grayscale intensity from camera is quantized between [0, 255],

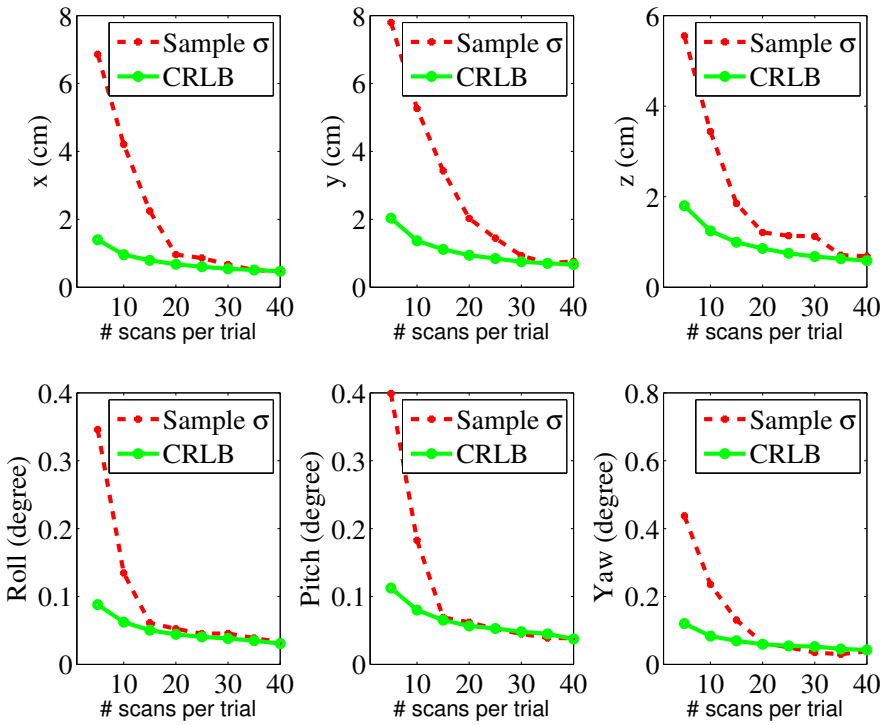
resulting into a large joint histogram ($256 \times 256 = 65,536$ bins) that needs to be estimated. The number of 3D points or observations (X_i, Y_i) of these random variables obtained from a single scan in the case of Velodyne data is typically of the order of 80,000. Therefore, if we use a single scan-image pair the joint histogram is largely under-sampled (Fig. 3.15(a)) because only about 80,000 observations are used to populate a histogram of 65,536 bins. However, if we use more data (i.e. scan-image pairs from multiple locations) to generate the joint histogram, it fills in the unobserved sections of the histogram (Fig. 3.15(b)). This results in a better estimate of the joint and marginal probability distributions of the random variables, which in turn increases the smoothness and convexity of the cost function (Fig. 3.15(d)). We can therefore use any gradient descent algorithm to quickly converge to the global optimum of this cost function. Fig. 3.16 shows the calibration results for when multiple scans are considered in the MI calculation. In particular, the experiments show that the standard deviation of the estimated parameters quickly decreases as the number of scans are increased by just a few. Here, the red plot shows the sample standard deviation (σ) of the calibration parameters computed over 1000 trials, where in each trial we randomly sampled $\{N = 5, 10, \cdots, 40\}$ scans from the available indoor and outdoor datasets to use in the MI calculation. The green plot shows the corresponding CRLB of the standard deviation of the estimated parameters.

In particular, we see that with as little as 10–15 scans, we can achieve very accurate performance. Moreover, we see that the sample variance asymptotically approaches the CRLB as the number of scans are increased, indicating this is an efficient estimator. In this experiment we took static snapshots of the laser scan and the camera image to avoid any errors due to motion of the vehicle. Although using the static snapshot is the best way to acquire data for calibration, if we have access to a good IMU mounted on the vehicle, the calibration process can be made even more user friendly. In that case we can motion-compensate the scan data using the IMU and then use it in the proposed calibration method. This allows for easy online calibration of the sensors without the need for acquiring static snapshots. We found that the calibration parameters obtained from the motion-compensated scans (using a good IMU) are close to those obtained from the static scans (Table 3.1).

3.4.2 Targetless Calibration: Performance with Different Initial Guess

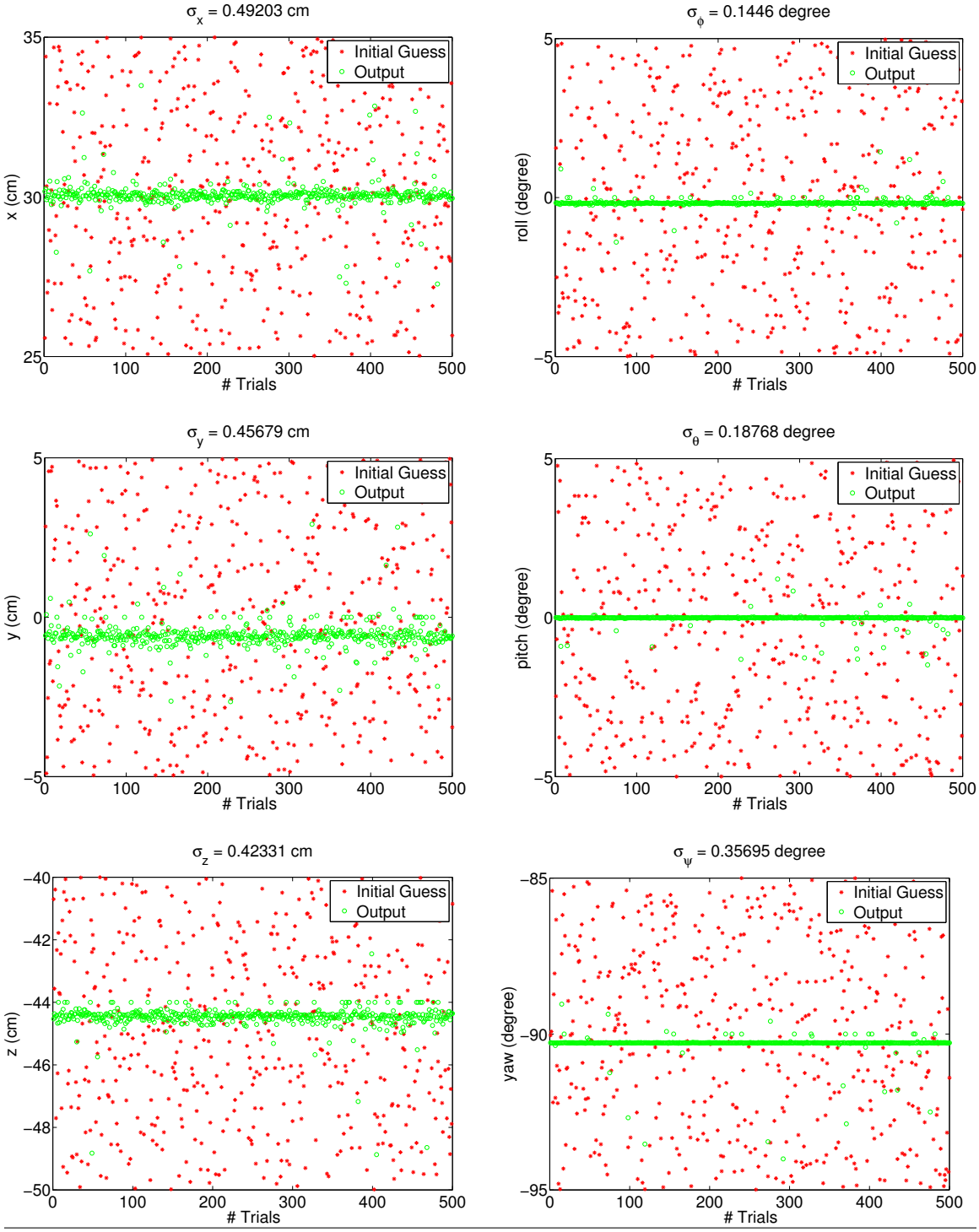
In this experiment we show the robustness of the proposed algorithm over the initial guess of the calibration parameters. As described in Algorithm 1, the proposed algorithm requires an initial guess of the calibration parameters that is generally calculated by manually measuring the distances and angles between the two sensors. Typically, the error in this measurement is of the order of 10 cm for translation parameters and 10° for rotation pa-

Figure 3.16 3D laser and omnidirectional camera multi-view calibration results. Here we use all five horizontal images from the Ladybug3 omnidirectional camera during the calibration. Plotted is the uncertainty of the recovered calibration parameters versus the number of scans used. The red (dashed line) plot shows the sample-based standard deviation (σ) of the estimated calibration parameters calculated over 1000 trials. The green (solid line) plot represents the corresponding CRLB of the standard deviation of the estimated parameters. Each point on the abscissa corresponds to the number of aggregated scans used per trial.



rameters. So, here we performed 500 independent trials with random initial guess (within the measurement errors) and observed that the algorithm converges to the correct calibration parameters (Fig. 3.17). In this experiment we used 20 randomly sampled scan-image pairs from our indoor and outdoor dataset. We observe that the standard deviation of the estimated translation parameters over these 500 trials is less than 0.5 cm and the standard deviation of the rotation parameters is less than $0.36°$. Therefore, this experiment clearly depicts the robustness of the proposed algorithm over a wide range of initial guessess of the calibration parameters that is within the acceptable range of manual errors.

Figure 3.17 Calibration performance for different initial conditions with 20 scan-image pairs. Here we perform 500 independent trials with random initial guess. The initial guess is marked in red and the output of the proposed calibration algorithm is marked in green.



3.4.3 Targetless Calibration: Computation Time Analysis

In this experiment we analysed the computation complexity of the proposed algorithm. In §3.4.1.3 we showed that as we increase the number of scans, from different view-points, the calibration performance increases. However, the increase in the number of scans also increases the computation time of the algorithm. Since the computation complexity of the algorithm is $O(n + m^2)$, where n is the number of 3D points used and m is the number of quantization bins of the random variables X and Y, if the number of quantization bins is fixed (here 256) then the computation time increases linearly with the increase in number of 3D points or scans. Fig. 3.18 shows a plot of computation time as a function of the number of scans used with a simple gradient descent algorithm [12] as the optimization method. We observe that the computation time (on a standard laptop with Intel Core i7-2670QM CPU @ 2.20 GHz) when the algorithm uses 20 scan-image pairs is of the order of 5 minutes, which we believe is acceptable for the calibration task. There is a clear trade-off between the computation time and the robustness of the algorithm as the increase in number of scans makes the algorithm more robust but it also increases the computation time. Since calibration is always an offline task there is no need for the algorithm to be real-time; however, we also do not want to wait for very long to obtain the results. Therefore, an optimal value of the number of scans should be chosen depending upon the application. In our experiments, we observed that 20 scans provide good calibration results (§3.4.1.3, §3.4.2) within 5 minutes, which we believe is acceptable for practical in-field operations of robots.

3.4.3.1 Targetless Calibration: Comparison with Available Ground-truth

The omnidirectional camera used in our experiments is pre-calibrated from the manufacturer. It has six 2-Megapixel cameras, with five cameras positioned in a horizontal ring and one positioned vertically, such that the rigid-body transformation of each camera with respect to a common coordinate frame, called the camera head (H), is well-known [82]. Here, $\mathbf{X}_{Hc_i}$ is the Smith et al. [153] coordinate frame notation, and represents the 6-DOF pose of the i^{th} camera (c_i) with respect to the camera head (H). Since we know $\mathbf{X}_{Hc_i}$ from the manufacturer, we can calculate the pose of the i^{th} camera with respect to the j^{th} camera as:

$$\mathbf{X}_{c_ic_j} = \ominus\mathbf{X}_{Hc_i} \oplus \mathbf{X}_{Hc_j}, \quad \{i \neq j\}. \tag{3.22}$$

In the previous experiments we used all 5 horizontally positioned cameras of the Ladybug3 omnidirectional camera system to calculate the MI; however, in this experiment we consider only one camera at a time and directly estimate the pose of the camera with

Figure 3.18 Computation time as a function of number of scans used for calibration. Computation time increases as the number of data points are increased. 1 scan contains approximately 80,000-100,000 3D points. More data results in better calibration performance, so there is a trade-off between computation time and the robustness of the algorithm.

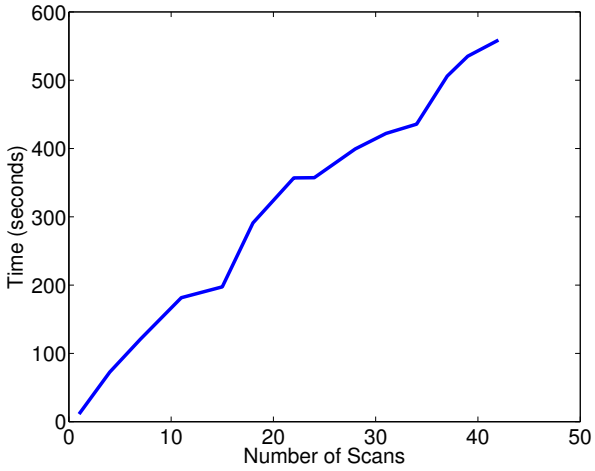
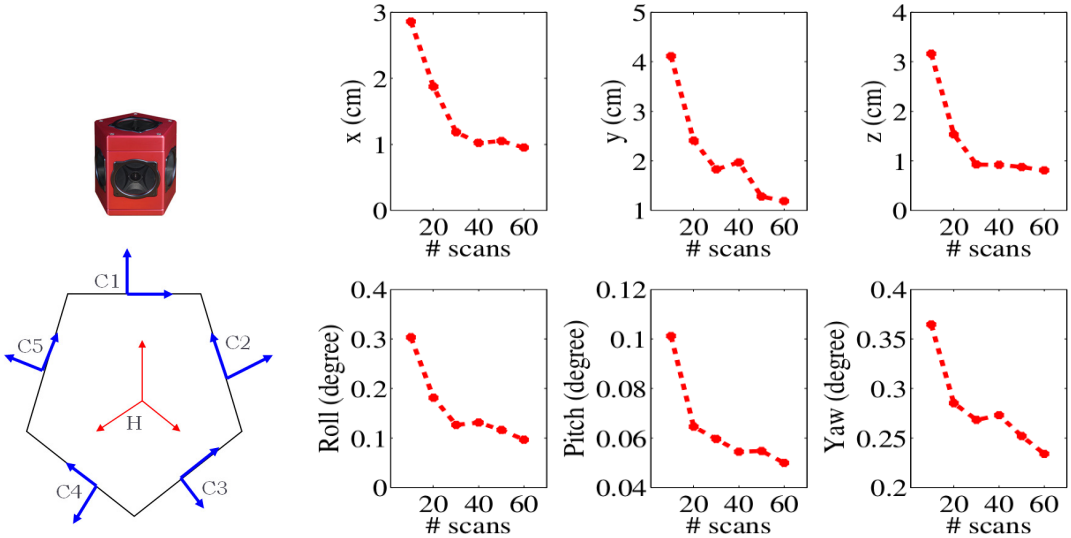


Figure 3.19 Comparison with manufacturer ground-truth. (a) A depiction of the coordinate frames corresponding to each camera (c_i) and camera head (H) of the Ladybug3 omnidirectional camera system. (b) Plotted are the mean absolute error in the relative-pose calibration parameters for the two side looking cameras (c_2 and c_5), i.e. $|\mathbf{X}_{c_2c_5} - \hat{\mathbf{X}}_{c_2c_5}|$, versus the number of scans used to estimate these parameters. The mean is calculated over 100 trials of sampling N scans per trial $\{N = 10, 20, \cdots, 60\}$. We see that the error decreases as the number of scans are increased.



(a) LB3 camera layout (b) Calibration error w.r.t. manufacturer provided values

respect to the laser reference frame ($\mathbf{X}_{\ell c_i}$). This allows us to calculate $\hat{\mathbf{X}}_{c_i c_j}$ from the estimated calibration parameters $\hat{\mathbf{X}}_{\ell c_i}$ and $\hat{\mathbf{X}}_{\ell c_j}$. Thus, we can compare the true value of $\mathbf{X}_{c_i c_j}$ (from the manufacturer data) with the estimated value $\hat{\mathbf{X}}_{c_i c_j}$. Fig. 3.19 shows one such comparison from the two side looking cameras of the Ladybug3 camera system. Here we see that the error in the estimated calibration parameters reduces with the increase in the number of scans and asymptotically approaches the expected value of the error (i.e., $\mathrm{E}[|\hat{\mathbf{\Theta}} - \mathbf{\Theta}|] \rightarrow 0$). It should be noted that in this experiment we used only a single camera as opposed to all 5 cameras of the omnidirectional camera system, thereby reducing the amount of data used in each trial by $1/5^{\text{th}}$. It is our conjecture that with additional trials, a statistically significant validation of unbiasedness could be achieved. Since the sample variance of the estimated parameters also approaches the CRLB as the number of scans are increased, in the limit our estimator should exhibit the properties of a minimum variance unbiased (MVUB) estimator (i.e., in the limiting case the CRLB can be considered as the true variance of the estimated parameters).

3.4.3.2 Comparison with Other Methods

We performed the following experiments to quantitatively benchmark results from our proposed methods against other published methods.

1. Comparison with Levinson and Thrun [91]: Levinson and Thrun [91] proposed an automatic calibration technique that uses correlation between depth discontinuities in the laser data and their projected edges in the corresponding camera images. In this experiment we replace our MI-based cost function with the criteria proposed by Levinson and Thrun:

$$\mathrm{LC}(X, Y; \mathbf{\Theta}) = \sum_{f=1}^{N} \sum_{p=1}^{|X^f|} X_p^f \cdot D_{i,j}^f, \tag{3.23}$$

where $\mathrm{LC}(\,\cdot\,)$ is Levinson's criteria for N scan image pairs, X_p^f is the depth discontinuity at the p^{th} point in scan f, and $D_{i,j}^f$ is the edge strength at projection of 3D point p onto the corresponding image f. The modified cost function can be written as:

$$\mathbf{\Theta} = \arg\max_{\mathbf{\Theta}} \mathrm{LC}(X, Y; \mathbf{\Theta}). \tag{3.24}$$

Fig. 3.20 shows a comparison of the proposed method with Levinson's method. In this experiment we used motion compensated scans captured in an outdoor urban environment (appendix C) to estimate the rigid-body transformation from both methods. In

Levinson's case only points corresponding to the edges of the surfaces are used, discarding a large amount of points corresponding to the ground plane and other flat surfaces present in the environment—therefore, it requires a relatively large number of scans and a structured calibration environment. Although, the plots show that for both methods the sample-based standard deviation of the estimated calibration parameters decreases as the number of scans are increased, the proposed method gives good calibration results with only 20 scans whereas Levinon's method requires nearly 100 scans to reach the same precision level. Unlike Levinson's method, our proposed method is whole-image based and uses *all* of the overlapping laser-image data. This allows our method to produce good calibration results with fewer scans even if the calibration environment is largely devoid of any linear depth discontinuities—the only criteria being that the scene have some distinctive reflectivity/intensity texture (e.g., a parking lot with painted parking stalls like in Fig. 3.7).

2. Comparison with Williams et al. [168]: In this experiment we replace the MI criteria with the χ^2 statistic used by Williams et al. [168]. The χ^2 statistic gives a measure of the statistical dependence of the two random variables in terms of the closeness of the observed joint distribution to the distribution obtained by assuming X and Y to be statistically independent:

$$\chi^2(X, Y; \mathbf{\Theta}) = \sum_{x \in X,\, y \in Y} \frac{\big(p(x, y; \mathbf{\Theta}) - p(x; \mathbf{\Theta}) p(y; \mathbf{\Theta})\big)^2}{p(x; \mathbf{\Theta}) p(y; \mathbf{\Theta})}. \tag{3.25}$$

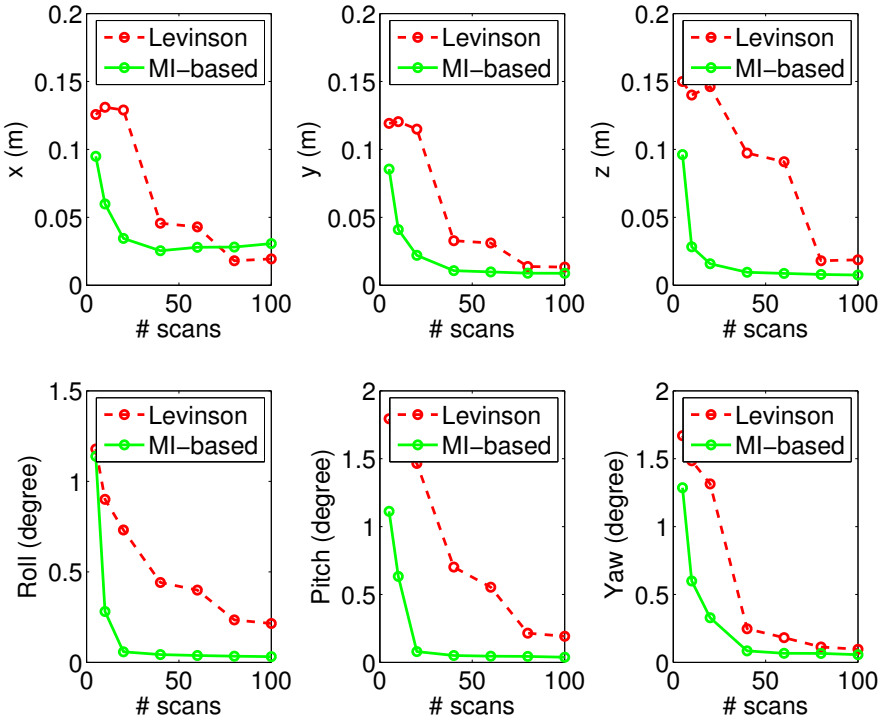
We can therefore modify the cost function given in (3.17) to:

$$\mathbf{\Theta} = \arg\max_{\mathbf{\Theta}} \chi^2(X, Y; \mathbf{\Theta}). \tag{3.26}$$

A comparison of the calibration results obtained from the χ^2 test (3.26) and the MI cost function (3.17) using 40 scan-image pairs is shown in Table 3.1. We see that the results obtained from the χ^2 statistics are similar to those obtained from the MI criteria. This is mainly because the χ^2 statistics and MI are equivalent and essentially capture the amount of correlation between the two random variables [102]. However, by aggregating several scans within a single optimization framework we generate a smooth cost function, which allows us to completely avoid the estimation of the initial guess of the calibration parameters, unlike Williams et al.

3. Comparison with target-based method: We compared the minimum variance results (i.e., estimated using 40 scans) of the targetless method with the target-based method and

Figure 3.20 Comparison with Levinson and Thrun [91]. Here we plot the uncertainty of the recovered calibration parameters versus the number of scans used. The red (dashed line) plot shows the sample-based standard deviation (σ) of the estimated calibration parameters calculated over 1000 trials using Levinson's method [91]. The green (solid line) plot shows the sample-based standard deviation of the estimated parameters using our proposed method. Each point on the abscissa corresponds to the number of aggregated scans used per trial. Clearly the proposed method converges to a good solution with significantly less number of scans.



found that they are very close (Table 3.1). The reprojection of 3D points onto the image using results obtained from these methods look very similar visually. Therefore, in the absence of ground truth, it is difficult to say which result is more accurate. The targetless method though, is definitely much faster and easier as it does not involve any manual intervention.

3.4.4 Time of Flight 3D Camera and Monocular Camera

In this section we present results from data collected from a 3D time-of-flight (TOF) camera [172] and a monocular camera [34] mounted on a horizontal bar (Fig. 3.21). A sample image obtained from the monocular camera is shown in Fig. 3.21(d) and the corre-

Table 3.1 Comparison of calibration parameters estimated by: MI-based targetless method with static scans, MI-based method with motion-compensated scans, feature alignment as reported in [91], χ^2 test as reported in [168], and checkerboard target-based method (§3.4.1.1).

	x [cm]	y [cm]	z [cm]	Roll [deg]	Pitch [deg]	Yaw [deg]
Targetless with static scans	30.5	-0.5	-42.6	-0.15	0.00	-90.27
Targetless with motion compensated scans	33.6	-0.7	-41.6	-0.20	-0.06	-90.14
Levinson and Thrun [91]	31.6	-0.3	-41.7	-0.05	0.05	-90.12
Williams et al. [168]	29.8	0.0	-43.4	-0.15	0.00	-90.32
Target-based	34.0	1.0	-41.6	0.01	-0.03	-90.25

sponding depth and intensity map of the scene obtained from the TOF 3D camera is shown in Fig. 3.21(b) and Fig. 3.21(c), respectively. The size of the depth map obtained from the 3D camera is 200×200 pixels, which equates to $40,000$ 3D points per scan.

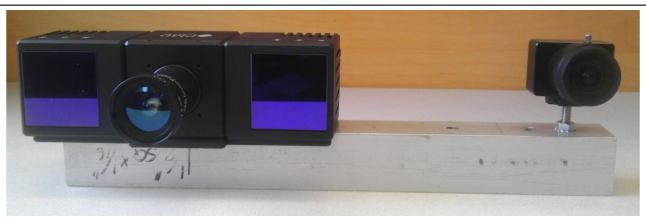
Targetless Calibration: In this experiment we use the 3D points with the intensity information along with the camera imagery to estimate the calibration parameters automatically without any specific targets within the MI-based framework. We assume that the intrinsic calibration parameters of the monocular camera are either known or are precomputed using any standard method (e.g., [174]). In Fig. 3.22 we show qualitative calibration results for projecting the 3D points onto the corresponding camera imagery using the estimated rigid-body transformation. We also show how the calibration results improve when multiple scans are considered in the MI-based calculation. We observe that the standard deviation of the estimated calibration parameters decreases and approaches the CRLB as the number of scans used to calculate the MI is increased.

3.4.5 2D Laser Scanner and Monocular Camera

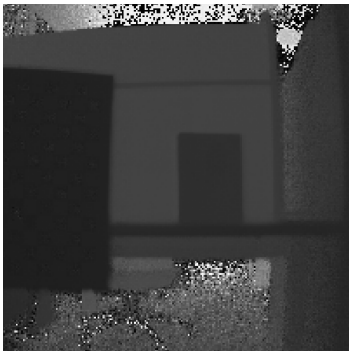
In this section we present results from data collected from a 2D laser scanner [62] and a monocular camera [34] mounted on a horizontal bar as shown in Fig. 3.23. This type of sensor setup is typical for an indoor SLAM problem.

Targetless Calibration: In this experiment we automatically calibrate the two sensors using the MI-based framework. In this case the single beam 2D laser scanner operates at 30 Hz and provided only 540 points per scan, of which only ~ 300 points overlapped with the camera imagery. So, the number of scans required to achieve a MVUB estimate of the calibration parameters was significantly large (of the order of few hundreds). Since

Figure 3.21 Data obtained from a 3D time-of-flight camera and monocular camera system.

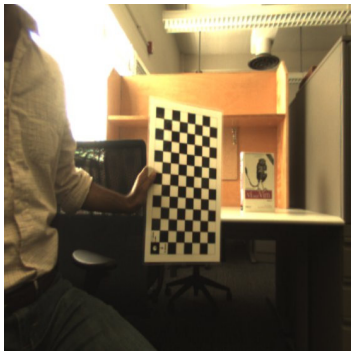


(a) A 3D TOF camera and a monocular camera setup



(b) Depth map from TOF camera

(c) Intensity map from TOF camera



(d) Image from monocular camera

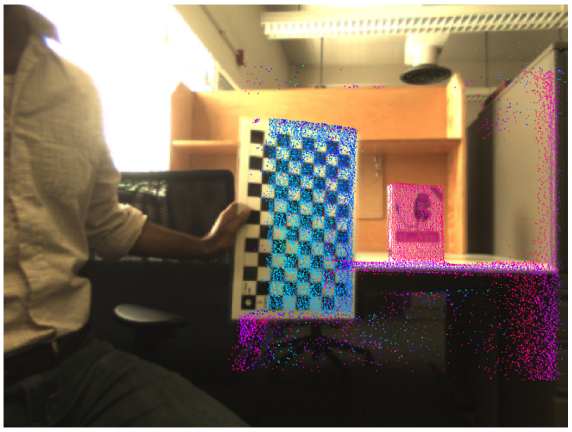
the scans can be arbitrary (i.e., there is no constraint on the calibration environment) it is not difficult to quickly capture a large number of scans from this setup. The quality of the MVUB estimate (calculated from 600 scans) is shown in Fig. 3.23(b). In Fig. 3.23(c) we plot the sample standard deviation of the estimated calibration parameters and the corresponding CRLB as a function of the number of scan pairs used. As observed in the earlier experiments (§3.4.1.3, §3.4.4), we see a decrease in parameter variance as the number of scans is increased and the sample standard deviation approaches that predicted by the CRLB.

3.5 Conclusion

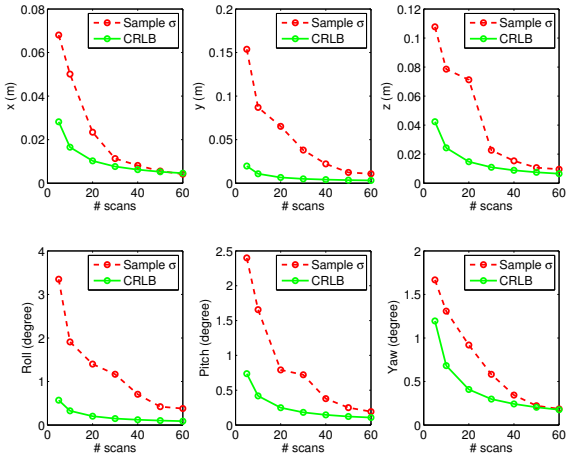
In this chapter we discussed the problem of extrinsic calibration of a laser range sensor and an optical camera system. Extrinsic calibration is the process of estimating the rigid-body transformation between the reference coordinate system of the two sensors. This rigid-body transformation allows reprojection of the 3D points from the range sensor coordinate frame to the 2D camera coordinate frame. Fusion of data provided by range and vision sensors can enhance various state-of-the-art computer vision and robotics algorithms.

Here, we presented two methods (*i*) target-based and (*ii*) targetless, to estimate the rigid-body transformation between a laser scanner and a camera system. The target-based

Figure 3.22 Results for the MI-based calibration of a 3D TOF camera and a monocular camera. (a) TOF point cloud projected onto the camera imagery; the points are color-coded based on distance from the camera. (b) We plot the uncertainty of the recovered calibration parameter versus the number of scans used. The red (dashed line) plot shows the sample-based standard deviation (σ) of the estimated calibration parameters calculated over 1000 trials. The green (solid line) plot represents the corresponding CRLB of the standard deviation of the estimated parameters. Each point on the abscissa corresponds to the number of aggregated scans used per trial. Note: The chekerboard pattern shown in the image is just for better visualization of the re-projected points and is not used for calibration.



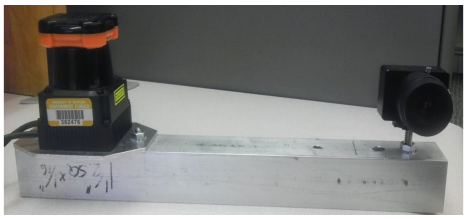
(a) TOF point cloud projected onto the camera imagery



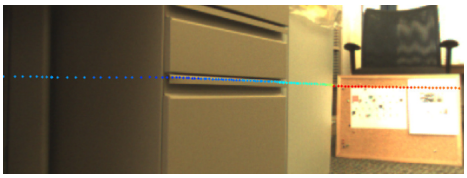
(b) MI-based calibration result

method minimally requires three views of a planar pattern visible from both the camera and the laser scanner. However, to get good results one needs several different views of large planar surfaces visible from the laser-camera system. The laser points lying on the planar surface and the normal of the plane (as estimated from the image data) provide a constraint on the rigid-body transformation between the two sensors. Although this method can provide a good estimate of the calibration parameters, it is laborious and time consuming. The targetless method, on the other hand, is completely data-driven and does not require any artificial targets to be placed in the field-of-view of the sensors. This makes the algorithm free from any degenerate configurations when calibrating the sensors unlike the target-based methods, which requires special attention to avoid any degeneracies in the data collected during calibration. Another drawback of the target-based method is it requires some kind of calibration setup. This is the reason why sensor calibration, when performed with the target-based technique, in a robotic application is typically performed once, and the same calibration is assumed to be true for rest of the life of that particular sensor suite. However, for robotics applications where the robot needs to go out into rough

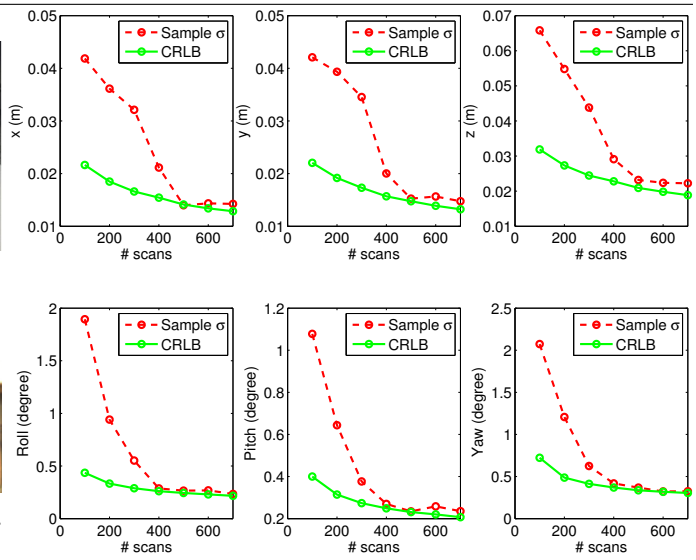
Figure 3.23 Results for the MI-based calibration of a 2D lidar and a monocular camera. (a) 2D laser scanner and a monocular camera mounted on a horizontal bar. (b) 2D lidar points projected onto camera image using the estimated transform. Points are color-coded based on distance from the camera: blue–close, red–far. (c) We plot the uncertainty of the recovered calibration parameter versus the number of scans used. The red (dashed line) plot shows the sample-based standard deviation (σ) of the estimated calibration parameters calculated over 1000 trials. The green (solid line) plot represents the corresponding CRLB of the standard deviation of the estimated parameters. Each point on the abscissa corresponds to the number of aggregated scans used per trial.



(a) 2D laser/camera setup



(b) 2D lidar points projected onto camera image



(c) Multi-view calibration result

terrain, assuming that the sensor calibration is not altered during a task is often not true. Although, we should calibrate the sensors before every task, it is typically not practical to do so if it requires to set up a calibration environment every time. The targetless method, being free from any such constraints, can be easily used to fine tune the calibration of the sensors *in situ*, which makes it applicable to in-field calibration scenarios.

The targetless method that we presented in this chapter utilizes the statistical dependence between the sensor measured surface intensity values. It is important to note that the reflectivity of the 3D points obtained from the range sensor and intensity of the pixel obtained from the camera are discrete signals generated by sampling the same physical scene, but in a different manner. Since the underlying structure generating these signals is common, they are statistically dependent upon each other. We use MI as the measure of this statistical dependence and formulate a cost function that is maximized for the correct calibration parameters. The source code of an implementation of the proposed algorithm in C++ is available for download from our server at `http://robots.engin.umich.edu/SoftwareData/ExtrinsicCalib`.

We showed that the targetless method works with a wide variety of sensors commonly used in indoor/outdoor robotics. Various experiments were performed to show the robustness and accuracy of the algorithm in typical robotics applications. Whether it is a 3D laser scanner and an omnidirectional camera system mounted on the roof of a car, or a 2D laser scanner and a monocular camera mounted on a robotic platform for indoor applications, the proposed method works equally well. Moreover, our algorithm also provides a measure of the uncertainty of the estimated parameters through the CRLB, which can be readily used within any probabilistic robotics perception framework.

CHAPTER IV

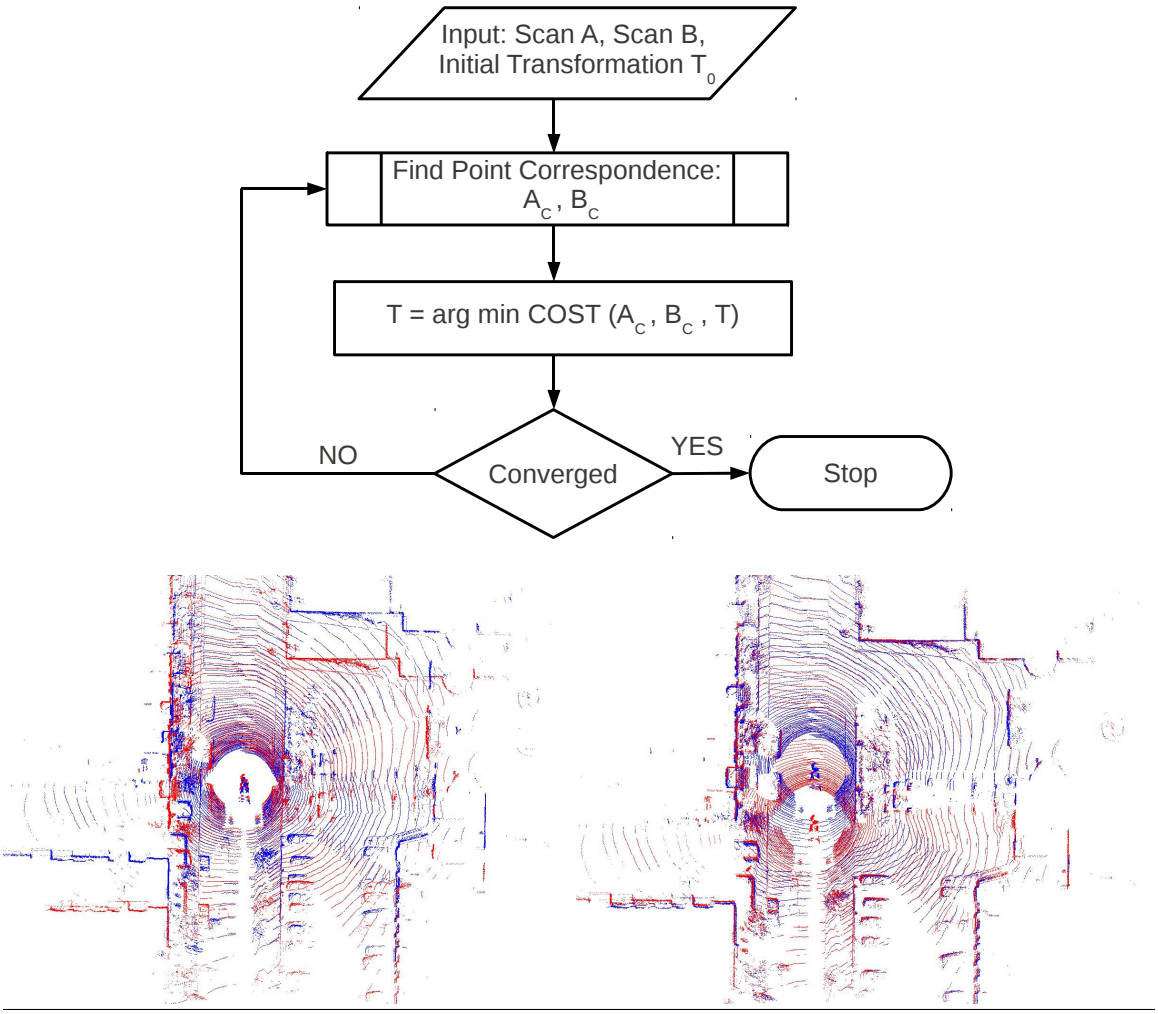
Alignment of Textured 3D Point Clouds

4.1 Introduction

In the previous chapter we described two techniques for extrinsic calibration of a laser scanner and a camera mounted on a robotic platform. These sensors help a robot to sense and understand the environment around them. The environment is sensed by these perception sensors, and knowledge about the environment is obtained by registering the current data with the previously perceived data. Substantial work has been done in registering 3D sensor data to obtain meaningful information about the environment as well as the current location of the robot. However, using fused 3D lidar and camera data for scan registration is yet not very common.

The problem of registering two 3D scans into a common reference frame has been researched for over three decades now. Arun et al. [9] presented a closed-form solution of this problem assuming that the point correspondence between the two sets of points is known. For a given set of corresponding points they proposed a least squares solution of the rigid-body transform [R, t], based on the singular value decomposition (SVD) of a 3×3 matrix. Horn [64] proposed an iterative solution to the same problem using quaternions to represent the rotation between the reference frames of the two point sets. Although the described methods provide a solution to the registration problem, they assume that the correspondence between the points is known. In most practical cases, this correspondence between the two different scans is not available directly. So, in order to solve the registration problem, point correspondence between the two scans needs to be established first. Several methods have been proposed to solve this problem over the last two decades, which can be broadly classified into the following categories: iterative methods, probabilistic methods and other methods.

Figure 4.1 Top: Flowchart of classical ICP framework. Bottom Left: Top-down view of two scans (Red/Blue) before alignment. Bottom Right: Scans (Red/Blue) after alignment using the ICP algorithm



4.1.1 Iterative methods

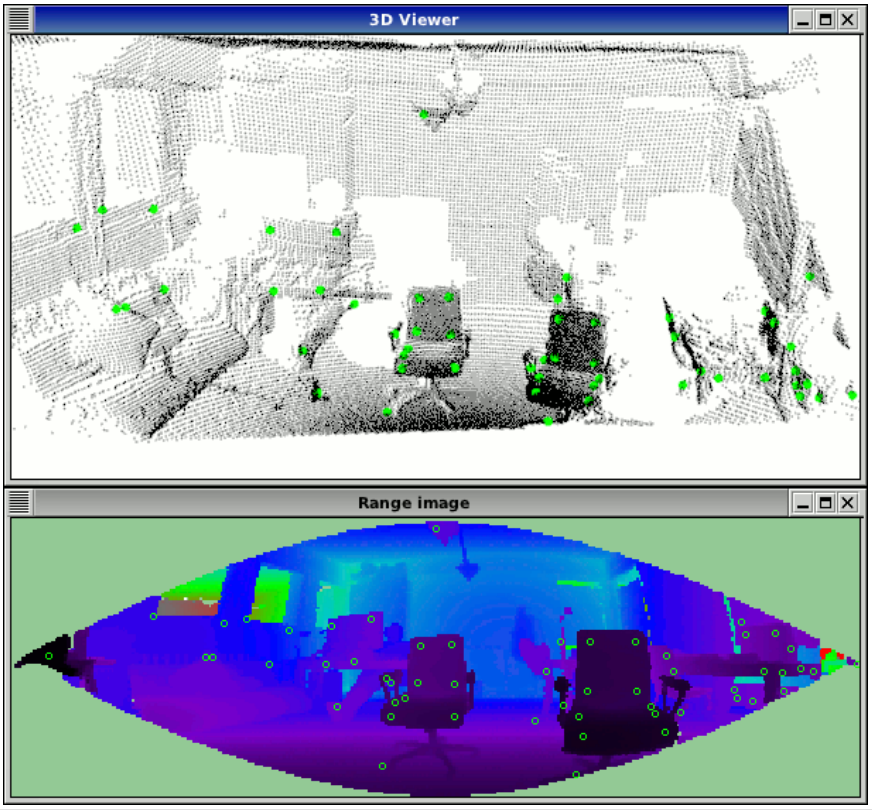
Iterative methods are one of the most common methods of scan matching, and have seen constant improvement since their introduction by Besl and McKay [17]. Besl and McKay proposed an iterative method for minimizing the Euclidean distance between corresponding points to obtain the relative transformation between the two scans. Their method is popularly known as the iterative closest point (ICP) algorithm and is widely used in solving the registration problem. Fig. 4.1 depicts the framework of the classical ICP algorithm. Chen and Medioni [25] introduced the point-to-plane variant of ICP owing to the fact that most of the range measurements are sampled from a locally planar surface of the environment. Since the introduction of ICP by Chen and Medioni [25] and Besl

and McKay [17], numerous variations of the algorithm has been proposed to increase the speed of the algorithm, improve the accuracy, and make it more robust to noise in the data [4, 26, 36, 39, 46, 109, 128]. A good survey of these variations of the classical ICP algorithm is presented by Rusinkiewicz and Levoy [139]. Most of the prior work on the alignment of a pair of scans assumes that the initial guess of the relative rigid-body transform is known. The initial guess may be estimated either by tracking the scanner position using an accelerometer or from the data itself. There are also methods that are completely data driven and use surface or 3D point cloud features to estimate the initial guess and then refine it using the ICP algorithms described above. In such methods local feature points and the corresponding feature descriptor are calculated in the two scans. These feature descriptors are then matched to establish correspondence, thereby estimating an initial guess of the relative rigid-body transform. Johnson and Hebert [69] introduced the novel *spin-images* as the local feature descriptors used for such an initial range scan alignment. Similarly Belongie and Malik [15] and Frome et al. [38] proposed a 2D and 3D shape context, respectively, for local feature matching. Ankerst et al. [6] introduced 3D shape histograms, based on partitioning the 3D space in which the object resides. Recently, Steder et al. [156] proposed a novel method for interest point detection and feature descriptor calculation in 3D range data called normal aligned radial feature (NARF) (Fig. 4.2). Rusu et al. [141] proposed a robust multi-dimensional feature descriptor called fast point feature histogram (FPFH), which is based on the local geometry around a point for 3D point cloud datasets. They showed that FPFH features can be computed online as the data becomes available from the sensors. The online computation of these features makes them suitable for registration of range data in real-time.

4.1.2 Probabilistic methods

One of the main reasons for the popularity of the ICP-based methods is their simplicity and speed. However, most of the deterministic algorithms discussed so far do not account for the fact that in real-world datasets, when the scans are coming from two different time instances, we never get exact point correspondences. Moreover, the scans are generally only partially overlapped and it is hard to establish point correspondences by applying a threshold on the point-to-point distance. Recently, several probabilistic techniques have been proposed that model the real-world data better than the deterministic methods. Granger and Pennec [45] introduced a general Maximum-Likelihood (ML) estimation of the transformation that aligns the two noisy point clouds, which they called Expectation Maximization Iterative Closest Point (EM-ICP). They showed that in the specific case of Gaussian noise, it corresponds to the ICP algorithm with Mahalanobis distance. Hahnel

Figure 4.2 Normal aligned radial feature (NARF) point extraction [156]. The 3D point cloud is first converted to a range image and the feature points are extracted from the estimated range image. The top panel shows the 3D point cloud and the corresponding NARF feature points ($green$ dots). The bottom panel shows the range image estimated from the 3D point cloud and the NARF feature points ($green$ circles).



and Burgard [49] applied ray tracing techniques to maximize the probability of alignment of two scans. Biber et al. [18] introduced an alternate representation of the range scans, the Normal Distribution Transforms (NDT), where they subdivide a 2D plane into cells and assign a normal distribution to each cell to model the distribution of points in that cell. They use this density to match the scans and, therefore, no explicit point correspondence is required (since they are matching the densities). Gruen and Akca [47] proposed a Least Squares 3D Surface Matching (LS3D) method, which estimates the 3D transformation parameters between two or more 3D surface patches, by minimizing the Euclidean distances between the surfaces. Montesano et al. [112] proposed probabilistic modeling of the ICP process that takes into account the uncertainty of the sensor location and the noises of the measurement process. Olson [123] describes a scan matching algorithm based upon the cross correlation of two lidar scans. Recently Segal et al. [146] proposed a generalized ICP (GICP) algorithm that is derived by attaching a probabilistic model to the cost function minimization step of the standard ICP algorithm. They assume that the 3D points

Figure 4.3 Local neighborhood structure of 3D points used in GICP algorithm [146] is captured by the sample covariance matrix. Planar surfaces show two dominant eigen-vectors of the sample covariance matrix (right). Uniformly distributed points show equal eigen-vectors in all three dimensions (left).

are samples from a Gaussian distribution, with covariance structure derived from the local neighborhood of the points in the environment. Thus, a point from wall, road, and other flat surfaces will have a covariance matrix with two dominant eigen-vectors, whereas the points sampled from irregular surfaces like bushes or trees will have a covariance matrix with equal eigen-vectors (Fig. 4.3).

4.1.3 Methods using color/intensity of the surface

In this section we discuss some methods that include visual cues from the camera imagery in the ICP framework. Incorporating visual information in the ICP framework has been suggested in several variants. Johnson and Kang [70] and Godin et al. [41] proposed a simple approach of incorporating color information in the ICP framework by augmenting the three color channels to the 3D coordinates of the point cloud. Although this technique adds the color information to the ICP framework, it fails to justify the mixing of 3D coordinates of a point and the RGB values from the color channel, as they are two entirely different entities. Recently Akca [2] proposed a novel method of using intensity information for scan matching. He proposed the concept of a quasi surface, which is generated by scaling the normal at a given 3D point by its color, and then matching the geometrical surface and the quasi surface in a combined estimation model. This approach works well when the environment is structured and the normal at a given point is easy to compute.

All of the aforementioned methods use the color information directly, i.e., they are using the very basic building blocks of the image data (RGB values), which does not provide strong distinction between the points of interest. However, there has been significant development over the last decade in the feature point detection and description algorithms employed by the computer vision and image processing community. We can now characterize any point in the image by high dimensional descriptors such as the scale invariant feature transform (SIFT) [96] or speeded up robust features (SURF) [13], as compared to

just RGB values alone. These high dimensional features provide a better measure of correspondence between points as compared to the Euclidean distance. The extrinsic calibration of 3D lidar and omnidirectional camera imagery allows us to associate these robust high dimensional feature descriptors to the 3D points. Once we have augmented the 3D point cloud with these high dimensional feature descriptors we can then use them to align the scans in a robust manner.

In this chapter we describe two methods of scan alignment that incorporate visual information obtained from camera imagery into the scan registration process. The first method describes a bootstrapping strategy that provides a good initial guess for the state-of-the-art GICP algorithm. In this method, point correspondences are established in the high dimensional feature space using the image-derived feature vectors and then these putative correspondences are used in a random sample consensus (RANSAC) [35] framework to obtain an initial rigid-body transformation that aligns the two scans. This initial transformation is then refined in a GICP framework as proposed by Segal et al. [146]. The second method, on the other hand, presents a novel MI-based algorithm that provides a robust framework for incorporating complementary information obtained from the camera and lidar modalities into the registration process directly.

The outline of the rest of the chapter is as follows: In section 4.2 we describe the visually bootstrapped generalized ICP (VB-GICP) algorithm. We divide the method into two parts, a RANSAC framework to obtain the initial transformation from SIFT correspondences and a refinement of this initial transformation via a GICP framework. In section 4.2.3 we present results showing the robustness of the bootstrapped method and present a comparison of this method with the unenhanced GICP algorithm. In section 4.3 we describe the Mutual Information (MI) framework for scan alignment. In section 4.4 we present results showing the robustness of the MI-based method and present a comparison of this method with GICP. Finally, in section 4.5 we summarize our findings.

4.2 Visually Bootstrapped Generalized ICP (VB-GICP)

In the previous chapter we presented two algorithms for the extrinsic calibration of a 3D laser scanner and an optical camera system. The extrinsic calibration of the two sensors allows us to project 3D points onto the corresponding omnidirectional image (and vice versa). This co-registration allows us to calculate high dimensional feature descriptors in the image (here we use SIFT) and associate them to a corresponding 3D lidar point that projects onto that pixel location (Fig. 4.4). Since only few 3D points are projected onto interesting parts of the image (i.e., where visual feature points are detected), only a

Figure 4.4 The co-registered camera and lidar data is shown below. The left panel shows the 3D points projected on the image plane, only points above the ground plane are shown here. The right panel shows SIFT features corresponding to the 3D points projected on the image plane. The points high up (e.g., on the facades of building) generally do not have a corresponding 3D point because of limited vertical field of view (FOV) of the laser scanner.

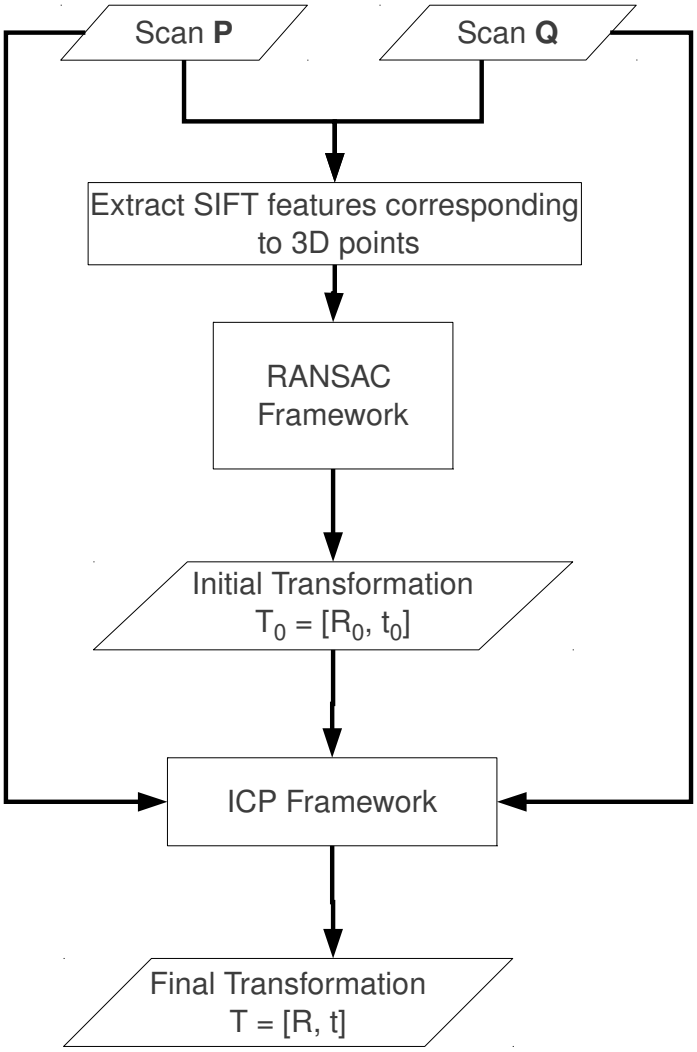
subset of the 3D points will have a feature descriptor assigned to them. Once we have augmented the 3D point cloud with the high dimensional feature descriptors, we then use them to align the scans in a two step process. In the first step, we establish putative point correspondence in the high dimensional feature space and then use these correspondences within a RANSAC framework to obtain a coarse initial alignment of the two scans. In the second step, we refine this coarse alignment using a generalized ICP framework [146]. Fig. 4.5 depicts an overview block-diagram of the proposed VB-GICP algorithm.

The novel aspect of the VB-GICP algorithm is in how we derive the initial coarse alignment. The initial alignment is intrinsically derived from the data itself using visual feature/lidar primitives available in the co-registered sensing modality. Note that initialization is typically the weakest link in any ICP-based methodology. By adopting the RANSAC framework, we are able to extend the convergence of generalized ICP over three times beyond the inter-scan distance that it normally breaks down. In the following, we explain our two-step algorithm in detail and discuss our novel concept of a camera consensus matrix (CCM).

4.2.1 RANSAC Framework

In the first part of our algorithm, we estimate a rigid-body transformation that approximately aligns the two scans using putative visual correspondences. We do so by matching the SIFT feature sets, S_P and S_Q, across the two scans and make the assumption that the matched feature points correspond to the same 3D point in Euclidean space. If we have three correct point correspondences, then we can calculate the rigid-body transformation

Figure 4.5 Block-diagram depicting the two step scan alignment process.



that aligns the two scans using the method proposed by Arun et al. [9]. However, if there exist outliers in the correspondences obtained by matching SIFT features, then this transformation will be wrong. Hence, we adopt a RANSAC framework [35] where we randomly sample three point correspondence pairs and iteratively compute the rigid-body transformation until we find enough consensus or exceed a preset maximum number of iterations based upon a probability of outliers.

The difficult aspect of this task is in establishing a good set of putative correspondences so as to get a sufficient number of inliers. Here we used the Point Grey Ladybug3 omnidirectional camera system [82]. The Ladybug3 has six 2-Megapixel (1600×1200) cameras, five positioned in a horizontal ring and one positioned vertically. Each sensor of the omnidirectional camera system has a minimally overlapping FOV as depicted in Fig. 4.6. The usable portion of the omnidirectional camera system essentially consists of five cameras

Figure 4.6 A depiction of the Ladybug3 omnidirectional camera system and a sample image showing the field of view of cameras 1 through 5.

spanning the 360° horizontal FOV. Unless we use prior knowledge on the vehicle's motion, we do not know *a priori* which camera pairs will overlap between the first and second scans. Hence, a simple global correspondence search over the entire omnidirectional image set will not give robust feature correspondence. Instead, in order to improve our putative feature matching, we exploit a novel camera consensus matrix concept that intrinsically captures the geometry of the omnidirectional camera system in order to establish a *geometrically consistent* set of putative point correspondences in SIFT space.

4.2.1.1 Camera Consensus Matrix

If the motion of the camera is known, then robustness to incorrect matches can be achieved by restricting the correspondence search to localized regions. Since we do not assume that we know the vehicle motion *a priori*, we first need to estimate these localized regions based upon visual similarity. To do so, we divide the FOV of the omnidirectional camera into n equally spaced regions. In our case we chose $n = 5$ because the five sensors of the omnidirectional camera naturally divide the FOV into five equi-spaced regions[1]. Once the FOV is partitioned we need to identify the cameras that have the maximum overlap between the two instances when the scans are captured. In our work, we assume that the motion of the vehicle is locally planar (albeit unknown).

[1]Note that in the case of catadioptric omnidirectional camera systems, the entire panoramic image can be divided into smaller equispaced regions.

For a small forward translational motion of the vehicle (Fig. 4.7) the maximum FOV overlap between scans **P** and **Q** occurs for the following pairs of cameras: {1-1, 2-2, 3-3, 4-4, 5-5}. Similarly, for large forward translational motion the maximum overlap of camera 1 of scan **P** can be with either of {1, 2 or 5} of scan **Q** (i.e., the forward looking cameras) (Fig. 4.7), whereas for the remaining four cameras of scan **P** the maximum overlap is obtained between {2-3, 3-3, 4-4, 5-4} of scan **Q**. This overlap of the cameras is captured in a matrix called the camera consensus matrix (CCM). The CCM is a [5 $\times$ 5] binary matrix where each element $\mathrm{C}(i,j)$ defines the correspondence consensus of the i^{th} camera of scan **P** with the j^{th} camera of scan **Q**, where 0 means no consensus and 1 means maximum consensus between the regions.

Similar to our translational motion example, we can also obtain the CCM for pure rotation of the vehicle about the yaw axis by circularly shifting the columns of the identity matrix as depicted in Fig. 4.8. Moreover, we can calculate the CCM matrices resulting from the combined rotational and translational motion of the vehicle by circularly shifting the CCM matrices from Fig. 4.7. Each resulting binary CCM represents a consistent geometry hypothesis of the camera motion and can be considered as a set of basis matrices spanning the entire space of possible CCMs arising due to the discrete planar vehicle motion assumed here. We vectorize these basis matrices by stacking the rows together into a vector, denoted $\mathbf{h}_i$, where each $\mathbf{h}_i$ corresponds to a valid geometry configuration CCM hypothesis.

4.2.1.2 Camera Constrained Correspondence Search

To use the concept of the CCM to guide our image feature matching, we first need to empirically compute the measured CCM arising from the visual similarity of the regions of scan **P** and scan **Q** using the available image data. Each element of the empirically derived CCM is computed as the sum of the inverse SIFT score (i.e., squared Euclidean distance) of the matches established between camera i of scan **P** and camera j of scan **Q**. This yields a measure of visual similarity between the two regions:

$$\tilde{\mathrm{C}}(i,j) = \sum_{k} 1/s_k, \tag{4.1}$$

where s_k is the SIFT matching score of the k^{th} match. This matrix is then normalized across the columns so that values are within the interval $[0,1]$ to comply with our notion that 0 means no consensus and 1 means maximum consensus:

$$\hat{\mathrm{C}}(i,j) = \tilde{\mathrm{C}}(i,j)/\max(\tilde{\mathrm{C}}(i)). \tag{4.2}$$

Figure 4.7 Top view of the omnidirectional camera system depicting the intersecting FOV of individual camera sensors as the omnidirectional camera-rig moves forward along the Y axis. For small translational motion (blue to red), the FOV of the cameras between scan **P** and scan **Q** does not change much, thereby giving maximal overlap with the same sensors and is described by the identity CCM matrix shown on the left. For large forward translational motion (blue to green), the FOV of the individual camera sensors does change and what was visible in camera 1 of scan **P** can now be visible in either of the forward looking cameras $\{1, 2 \text{ or } 5\}$ of scan **Q**, resulting in the sample CCM matrices shown on the right.

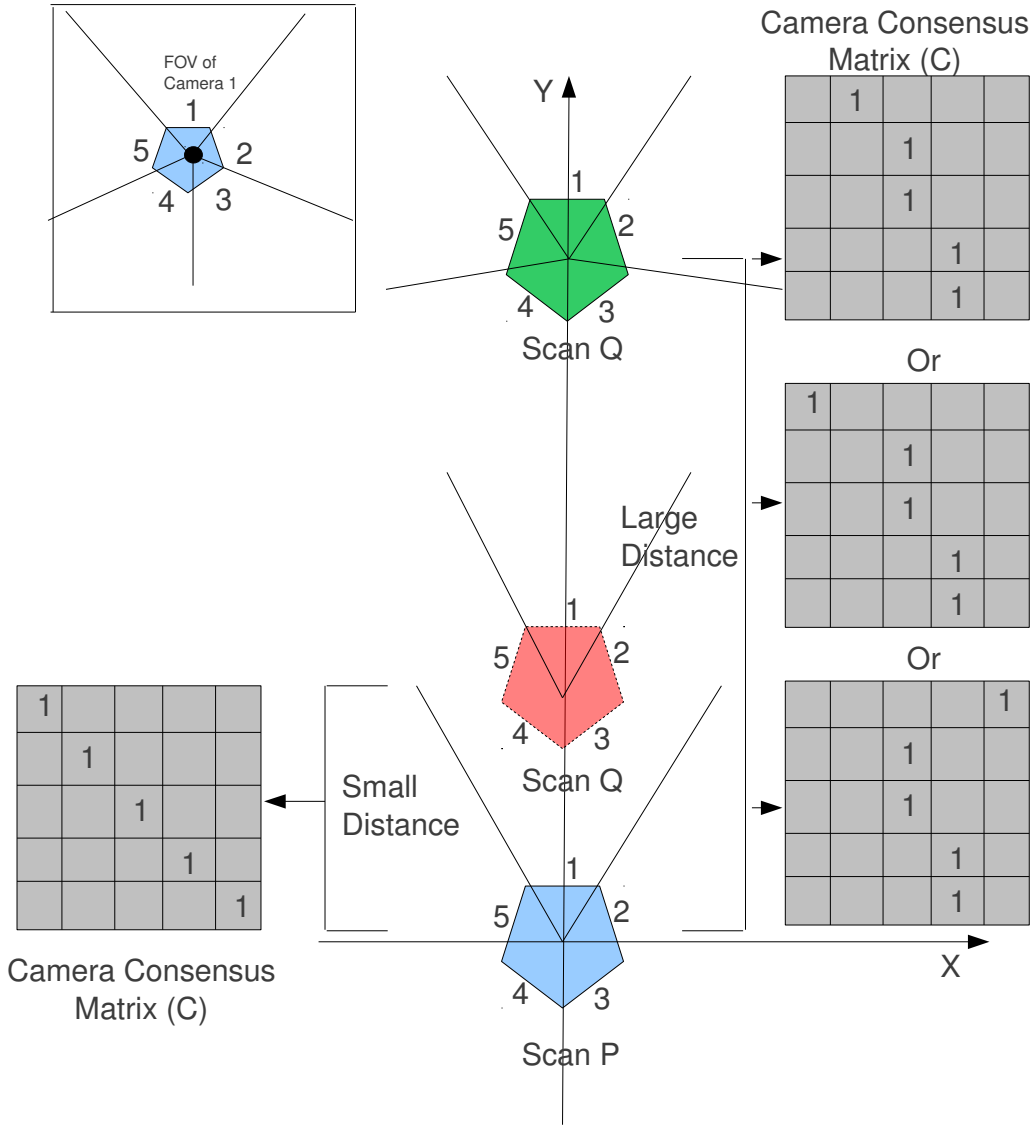
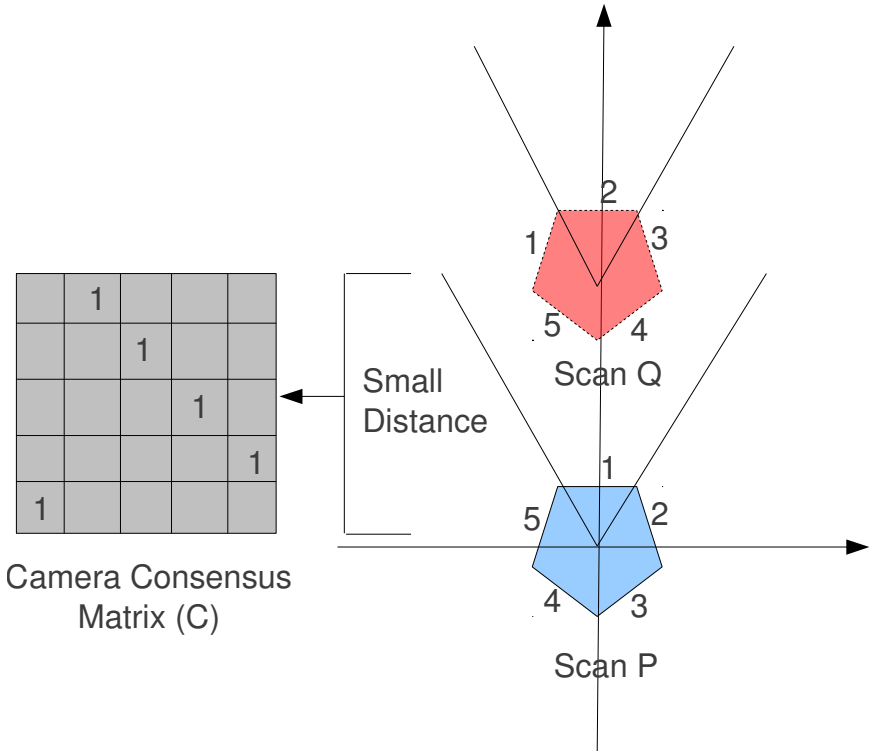


Figure 4.8 Top view of the omnidirectional camera system depicting the intersecting FOV of individual camera sensors as the camera-rig rotates about the yaw axis. Here, we have shown one possible discrete rotation such that the FOV of each sensor is circularly shifted by one unit, resulting in the sample CCM shown on the left. In this case, five such discrete rotations are possible.



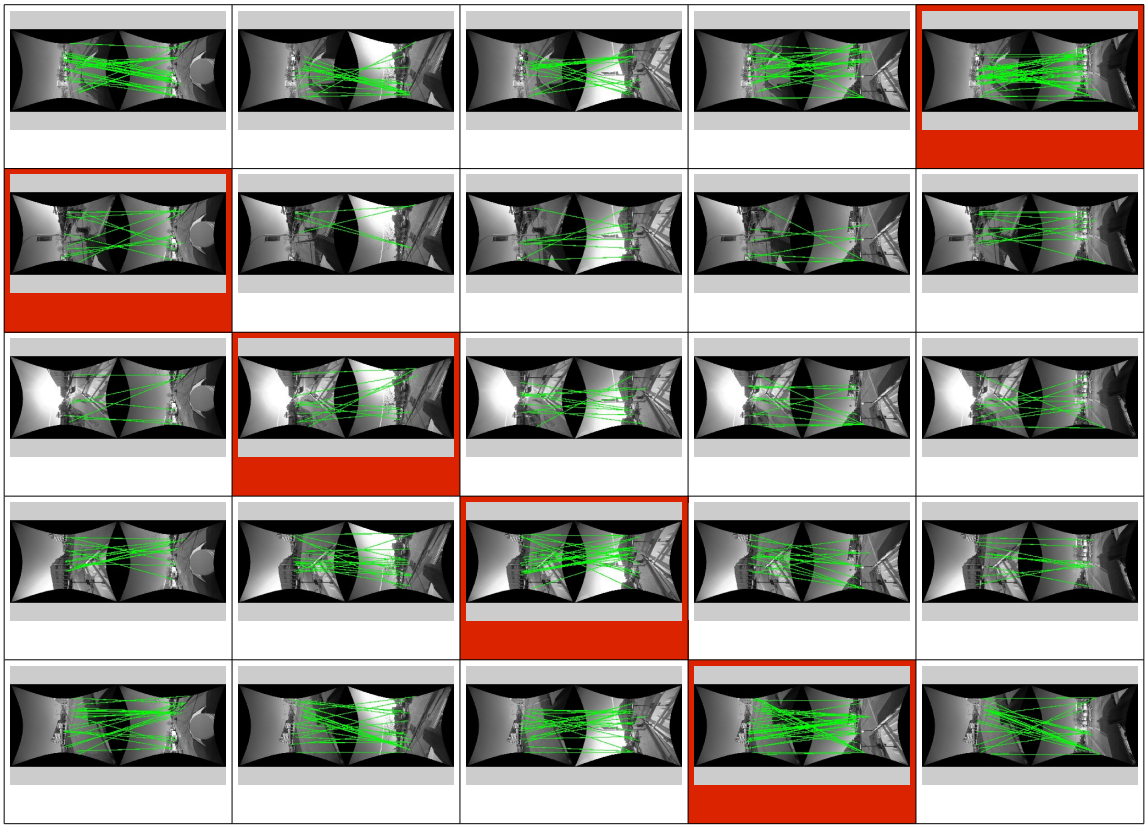
Here, $\max(\tilde{\mathrm{C}}(i))$ denotes the maximum value in the i^{th} row of the matrix $\tilde{\mathrm{C}}$.

This matrix $\hat{\mathrm{C}}$ is then vectorized to obtain the corresponding camera consensus vector $\hat{\mathbf{c}}$. To determine which ideal CCM hypothesis is most likely, we project this vector to all the hypothesis basis vectors $\mathbf{h}_i$ and calculate the orthogonal error of projection:

$$e_i = \|\hat{\mathbf{c}} - \mathbf{h}_i \frac{\hat{\mathbf{c}} \cdot \mathbf{h}_i}{\|\hat{\mathbf{c}}\| \|\mathbf{h}_i\|}\| \tag{4.3}$$

The basis vector $\mathbf{h}_i$ that has the least orthogonal error of projection yields the closest hypothesis on the CCM. This geometrically consistent camera configuration is then used for calculating the camera constrained SIFT features. Fig. 4.9 depicts a typical situation where the CCM yields a more robust feature correspondence as compared to the simple global correspondence search alone. The CCM-consistent putative correspondences are then used in the RANSAC framework to estimate the rigid-body transformation that aligns the two scans. The complete RANSAC algorithm to estimate the rigid-body transformation is outlined in Algorithm 2.

Figure 4.9 This figure shows the pairwise exhaustive SIFT matches obtained across the five cameras of scan **P** and scan **Q**. The corresponding empirically measured CCM is shown below on the left, and the closest matching binary CCM hypothesis is shown below on the right. The blocks highlighted in red indicate the CCM-consistent maximal overlap regions. In this case, the resulting CCM hypothesis indicates a clockwise rotational motion by one camera to the right (refer to Fig. 4.8).



0.7386	0.3669	0.2658	0.5588	1
1	0.6307	0.3590	0.4303	0.4800
0.6266	1	0.5160	0.7388	0.7482
0.4475	0.5219	1	0.5846	0.2976
0.6395	0.3719	0.2865	0.6600	1

Camera Consensus Matrix (CCM) based on visual similarity

0	0	0	0	1
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0

Closest hypothesis corresponding to the CCM

Algorithm 2 RANSAC Framework

1: **Input:** Scans **P** and **Q** with SIFT features S_P, S_Q.
2: **Output:** The estimated transformation $[\mathrm{R}, \mathbf{t}]$.
3: Establish camera-constrained SIFT correspondences between S_P and S_Q (camera constrained correspondence search (CCCS)).
4: Store the matches in a list L.
5: **while** iter $<$ MAXITER **do**
6: Randomly pick 3 pairs of points from the list L.
7: Retrieve these 3 pair of points from **P** and **Q**.
8: Calculate the 6-DOF rigid-body transformation $[\mathrm{R}, \mathbf{t}]$ that best aligns these 3 points.
9: Store this transformation in an array M, $M[iter] = [\mathrm{R}, \mathbf{t}]$
10: Apply the transformation to 3D points in **Q** to map Scan **Q**'s points into the reference frame of Scan **P**: $\mathbf{q}'_i = \mathrm{R}\mathbf{p}_i + \mathbf{t}$
11: Calculate the set cardinality of pose-consistent SIFT correspondences that agree with the current transformation (i.e., those that satisfy a Euclidean threshold on spatial proximity): $n = |(\mathbf{Q}'(L) - \mathbf{P}(L)) < \epsilon|$
12: Store the number of pose-consistent correspondences in an array N, $N[iter] = n$
13: $iter = iter + 1$
14: **end while**
15: Find the index i that has maximum number of correspondences in N.
16: Retrieve the transformation corresponding to index i from M. $[\mathrm{R}, \mathbf{t}]$ = $M[i]$. This is the required transformation.

4.2.2 ICP Framework

Our method to refine the initial transformation obtained from section 4.2.1 is based upon the GICP algorithm proposed by Segal et al [146]. The GICP algorithm is derived by attaching a probabilistic model to the cost function minimization step of the standard ICP algorithm outlined in Algorithm 3. In this section we review the GICP algorithm as originally described in [146].

The cost function at line 13 of the standard ICP algorithm (Algorithm 3) is modified in [146] to give the generalized ICP algorithm. In GICP the point correspondences are established by considering the Euclidean distance between the two point clouds **P** and **Q**. Once the point correspondences are established, the ICP cost function is formulated as a maximum likelihood estimate (MLE) of the transformation "T = [R, $\mathbf{t}$]" that best aligns the two scans.

In the GICP framework the points in the two scans are assumed to be coming from Gaussian distributions, $\hat{\mathbf{P}}_i \sim \mathcal{N}(\mathbf{p}_i; \mathrm{C}_i^P)$ and $\hat{\mathbf{q}}_i \sim \mathcal{N}(\mathbf{q}_i; \mathrm{C}_i^Q)$, where $\mathbf{p}_i$ and $\mathbf{q}_i$ are the mean or actual points and C_i^P and C_i^Q are sample-based covariance matrices associated with the measured points. Now in the case of perfect correspondences (i.e., geometrically

Algorithm 3 Standard ICP Algorithm [146]

Input: Two point clouds: **P**, **Q**;
An initial transformation: $[\mathrm{R}_0, \mathbf{t}_0]$.
Output: The correct transformation, $[\mathrm{R}, \mathbf{t}]$, which aligns **P** and **Q**.
$[\mathrm{R}, \mathbf{t}] \leftarrow [\mathrm{R}_0, \mathbf{t}_0]$
while not converged **do**
  for $i \leftarrow 1$ to N **do**
    $\mathbf{q}_i \leftarrow FindClosestPointInQ(\mathrm{R}\mathbf{p}_i + \mathbf{t})$
    if $\|\mathbf{q}_i - (\mathrm{R}\mathbf{p}_i + \mathbf{t})\| <= d_{max}$ **then**
      $w_i \leftarrow 1$;
    else
      $w_i \leftarrow 0$;
    end if
  end for
  $[\mathrm{R}, \mathbf{t}] \leftarrow \mathrm{argmin}_{[\mathrm{R},\mathbf{t}]} \sum_i w_i \|(\mathrm{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2$
end while

consistent with no errors due to occlusion or sampling) and correct transformation, $\mathrm{T}^* = [\mathrm{R}^*, \mathbf{t}^*]$:

$$\mathbf{q}_i = \mathrm{R}^*\mathbf{p}_i + \mathbf{t}^*. \tag{4.4}$$

But for an arbitrary transformation $\mathrm{T} = [\mathrm{R}, \mathbf{t}]$, and noisy measurements $\mathbf{p}_i$ and $\mathbf{q}_i$, the alignment error can be defined as $\mathbf{d}_i = \mathbf{q}_i - (\mathrm{R}\mathbf{p}_i + \mathbf{t})$. Therefore, the ideal distribution from which $\mathbf{d}_i^{(\mathrm{T}^*)}$ is drawn is given as:

$$\begin{aligned} \mathbf{d}_i^{(\mathrm{T}^*)} &\sim \mathcal{N}(\mathbf{q}_i - (\mathrm{R}^*\mathbf{p}_i + \mathbf{t}^*), \mathrm{C}_i^Q + \mathrm{T}^*\mathrm{C}_i^P\mathrm{T}^{*\top}) \\ &= \mathcal{N}(\mathbf{0}, \mathrm{C}_i^Q + \mathrm{T}^*\mathrm{C}_i^P\mathrm{T}^{*\top}). \end{aligned}$$

Here $\mathbf{p}_i$ and $\mathbf{q}_i$ are assumed to be drawn from independent Gaussians. Thus, the required transformation T is the MLE computed by setting:

$$\mathrm{T} = \underset{\mathrm{T}}{\mathrm{argmax}} \prod_i p(\mathbf{d}_i^{(\mathrm{T}^*)}) = \underset{\mathrm{T}}{\mathrm{argmax}} \sum_i \log p(\mathbf{d}_i^{(\mathrm{T}^*)}), \tag{4.5}$$

which can be simplified to:

$$\mathrm{T} = \underset{\mathrm{T}}{\mathrm{argmin}} \sum_i \mathbf{d}_i^{\mathrm{T}} (\mathrm{C}_i^Q + \mathrm{T}\mathrm{C}_i^P\mathrm{T}^T)^{-1}\mathbf{d}_i. \tag{4.6}$$

The rigid-body transformation T given in (4.6) is the MLE refined transformation that best aligns scan **P** and scan **Q**.

4.2.3 Experiments and Results

We present results from real data collected from a 3D laser scanner (Velodyne HDL-64E) and an omnidirectional camera system (Point Grey Ladybug3) mounted on the roof of a Ford F-250 vehicle (Appendix C). We use the pose information available from a high-end inertial measurement unit (IMU) (Applanix POS-LV) as the ground truth to compare the scan alignment errors. We performed the following experiments to analyze the robustness of the algorithm.

4.2.3.1 Experiment 1

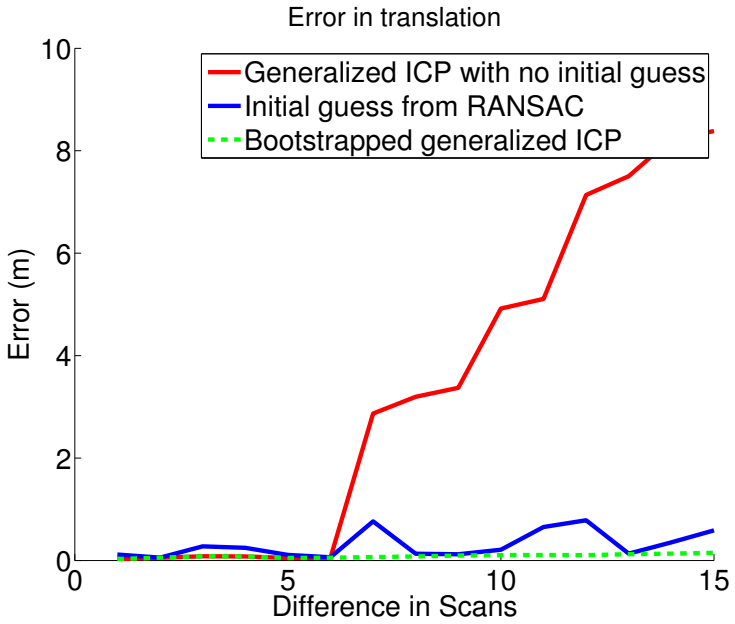
In the first experiment we selected a series of 15 consecutive scans captured by the laser-camera system in an outdoor urban environment collected while driving around downtown Dearborn, Michigan at a vehicle speed of approximately $15.6\ \mathrm{m/s}$ ($35\ \mathrm{mph}$). The average distance between the consecutive scans is approximately $0.5\ \mathrm{m}$–$1.0\ \mathrm{m}$. In this experiment we fixed the first scan to be the reference scan and then tried to align the remaining scans (2–15) with the base scan using *(i)* the GICP alone, *(ii)* our RANSAC initialization alone, and *(iii)* the bootstrapped generalized ICP algorithm seeded by our RANSAC solution (VB-GICP). The error in translational motion between the base scan and the remaining scans obtained from these algorithms is plotted in Fig. 4.10. We found the plotted error trend to be typical across all of our experiments—in general the GICP algorithm alone would fail after approximately 5 or so scans of displacement when not fed an initial guess. However, by using our VB-GICP algorithm, we were able to significantly extend GICP's convergence out past 15 scans of displacement.

We repeated this experiment for 10 sets of 15-scan pairs (i.e., 150 scans in total) from different locations in Dearborn and calculated the average translational and rotational error as a function of the intra-scan displacement. The resulting error statistics are tabulated in Table 4.1 where we see that the VB-GICP is able to provide sub $25\ \mathrm{cm}$ translational error at 15 scans apart, while GICP alone begins to fail after only 5 scans of displacement.

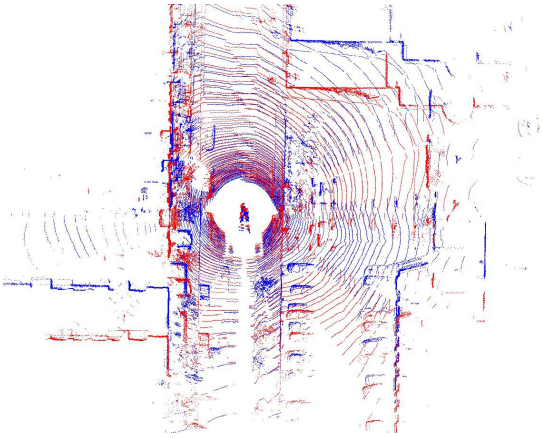
4.2.3.2 Experiment 2

In the second experiment, we compared the output of GICP and our VB-GICP in a real-world application-driven context. For this experiment we drove a $1.6\ \mathrm{km}$ loop around downtown Dearborn, Michigan with the intent of characterizing each algorithm's ability to serve as a registration engine for localizing and 3D map building in an outdoor urban environment. For this purpose we used a pose-graph simultaneous localization and mapping (SLAM) framework where the ICP-derived pose constraints served as edges in

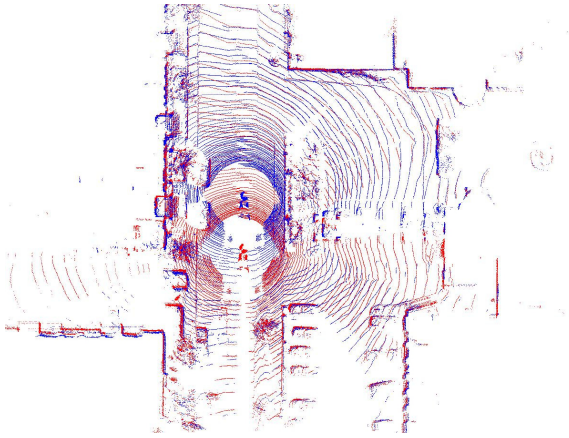
Figure 4.10 Graph showing the error (a) in translation as the distance between scans A and B is increased. Top view of the 3D scans aligned with the output of GICP (b) and VB-GICP (c) for two scans that are 10 time steps apart. Note that the GICP algorithm fails to align the two scans when unaided by our novel RANSAC initialization step.



(a) Error comparison between GICP and VB-GICP.



(b) GICP for Scan 10



(c) VB-GICP for Scan 10

Table 4.1 This table summarizes the error in scan alignment. We show here the translation and rotational error between scan pairs {1-2, 1-5, 1-10, 1-15} obtained at different locations. The average error shown below is computed over 100 trials. Here we have used the pose of the vehicle obtained from a high-end IMU as ground-truth to calculate all the errors. The alignment error for GICP quickly increases as the separation between the scans is increased. GICP starts to fail after only 5 scans of displacement ($3-5$ m), whereas VB-GICP provides better convergence even beyond 10 scans of displacement ($8-10$ m).

	Generalized ICP with no initial guess						Initial guess from RANSAC						Bootstrapped generalized ICP					
Scans	T (m)		Ax (degrees)		An (degrees)		T (m)		Ax (degrees)		An (degrees)		T (m)		Ax (degrees)		An (degrees)	
	Err	Std	Err	Std	Err	Std	Err	Std	Err	Std	Err	Std	Err	Std	Err	Std	Err	Std
1-2	.047	.011	0	0	.05	.02	.15	.02	0	0	.223	.0003	.04	.010	0	0	.057	.110
1-5	.546	.173	.570	.20	1.15	.344	.20	.03	.43	.15	.230	.0001	.084	.010	.025	.090	.058	.006
1-10	6.37	.868	.710	.25	1.72	.573	.51	.09	.59	.01	.745	.0044	.145	.015	.030	.010	.057	.012
1-15	10.34	.834	1.86	.13	2.86	.057	1.02	.02	1.35	.54	1.15	.0021	.220	.008	.042	.015	.070	.017

T = Error in translation (meters); Ax = Error in rotation axis (degrees); An = Error in rotation angle (degrees)
Err = Average Error; Std = Standard Deviation

the graph. We employed the open-source incremental smoothing and mapping (iSAM) algorithm by Kaess [72] for inference. In our experiment the pose-constraints are obtained only from the scan matching algorithm and no odometry information is used in the graph.

Fig. 4.11 shows the vehicle trajectory given by the iSAM algorithm (green) overlaid on top of OmniStar HP global positioning system (GPS) data ($\sim$2 cm error) for ground-truth (red). Here the pose constraints were obtained by aligning every third scan using GICP with no initial guess from odometry. As we can see in Fig. 4.11(b), the resulting iSAM output differs greatly from the ground-truth. This mainly occurs because the generalized ICP algorithm does not converge to the global minimum when it is initialized with a poor guess, which means the pose-constraints that we get are biased, and hence a poor input to iSAM. Fig. 4.11(d) shows the resulting vehicle trajectory for our VB-GICP algorithm when given as input to the iSAM algorithm, which agrees well with the GPS ground-truth.

4.3 Mutual Information based Alignment

In the previous section we presented a method for bootstrapping the 3D point cloud based scan registration algorithm using visual data from camera imagery. We utilized the sensor calibration to project 3D points from lidar onto the corresponding image (and vice versa), thereby extracting high-dimensional feature descriptors from the image (SIFT [96], SURF [13], etc.) and associated them to a corresponding 3D lidar point that projects onto

Figure 4.11 iSAM output with input pose constraints coming from GICP and VB-GICP. Here, the red trajectory is the ground-truth coming from GPS and the green trajectory is the output of the iSAM algorithm. The *start* and *end* point of the trajectory are the same and is denoted by the black dot.

(a) iSAM with GICP open-loop.

(b) iSAM with GICP closed-loop.

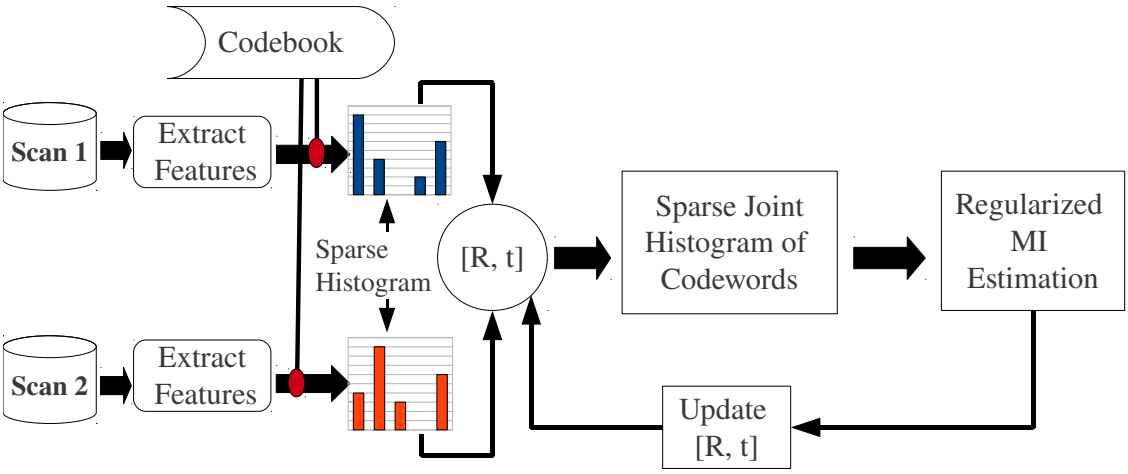
(c) iSAM with VB-GICP open-loop.

(d) iSAM with VB-GICP closed-loop.

that pixel location. Incorporating visual information from this co-registered omnidirectional camera imagery allowed us to provide a good initial guess and a more accurate set of point correspondences to the GICP algorithm by taking advantage of high dimensional image feature descriptors. Therefore, in the absence of a good initial guess (e.g., from odometry) the proposed VB-GICP algorithm provided a robust framework for scan alignment by estimating an initial guess on the rigid-body transformation from the data itself. Although, the proposed VB-GICP algorithm is robust and gives accurate results even when the overlap between the two scans is significantly less, it is still a loosely-coupled way of utilizing camera imagery. Moreover, the number of iterations of the RANSAC step required to obtain a good initial guess can be significantly large, thereby making the overall algorithm not suitable for real-time applications. Moreover, the VB-GICP algorithm is also dependent upon the omni-directional camera geometry to obtain good putative correspondences (via CCCS) in the RANSAC step.

In this section we describe a novel MI-based scan registration algorithm that allows for the principled fusion of camera and lidar modality information within a single optimization

Figure 4.12 The proposed MI-based scan registration method learns a codebook of the high-dimensional features extracted from the scans. Using this codebook the empirical histograms of codewords present in the scans are computed for a given rigid-body transformation. The MI is optimally estimated from them using a James-Stein-type shrinkage estimator. This MI is maximized at the sought after transformation parameters that aligns the two scans.



framework (tightly-coupled). Unlike VB-GICP, the MI-based algorithm is independent of the geometry of the lidar-camera system used. As described in the previous section we first extract high-dimensional features from the camera imagery and associate them to the corresponding points in the lidar data. Moreover, we also extract 3D features (FPFH [141], rotation invariant feature transform (RIFT) [83], spin-images [69], etc.) from the point cloud and combine the camera and lidar derived features to form a robust high-dimensional feature vector, which can be calculated at some keypoints of the scan. This allows us to represent a scan as a collection of high-dimensional feature vectors. Thus, for any two overlapping scans the joint distribution of these features should show maximum correlation when viewed under the correct rigid-body transformation. Therefore, here we use concepts from statistics and information theory to formulate a MI-based cost function to solve the scan registration problem. An overview of this method is shown in Fig. 4.12.

4.3.1 Mathematical formulation

We first create a dictionary of *codewords* representing the quantization of the high-dimensional features extracted in the scans. We extract N such features (training samples) from a set of scans called the training dataset (Fig. 4.13). We use a hierarchical k-means

Figure 4.13 The codebook and target distribution are learned from the training dataset, and all experiments are performed on the testing dataset. It should be noted that the training and testing datasets are captured in similar outdoor urban environments, though not the same. It is important for the codebook to be representative, but the testing and training environments need not be identical.

(a) Sample images from training dataset (*Ford Campus*)

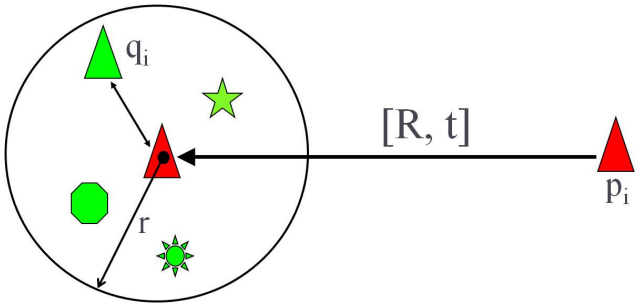
(b) Sample images from testing dataset (*Downtown*)

clustering [120] algorithm on the training samples to cluster the feature space into K clusters. The centroids of these clusters are defined as *codewords* $\{c_i; i = 1, 2, \cdots, K\}$ and the collection of these codewords is called the *codebook*. We use this codebook to map any feature vector to a unique integer i corresponding to the codeword c_i that gives a maximum similarity score with the feature vector.

We consider the collection of these codewords present in a scan as the random variables X and Y. The marginal and joint probabilities of these random variables $p_X(x)$, $p_Y(y)$ and $p_{XY}(x, y)$ can be obtained from the normalized marginal and joint histograms of the codewords present in the scans that we want to align. Let **P** and **Q** be the two scans that we want to align. Let $C^P = \{c_i^p; i = 1, \cdots, n\}$ and $C^Q = \{c_i^q; i = 1, \cdots, m\}$ be the set of codewords, and $\{\mathbf{p}_i; i = 1, \cdots, n\}$ and $\{\mathbf{q}_i; i = 1, \cdots, m\}$ be the set of 3D points corresponding to the codewords present in scans **P** and **Q**, respectively. If the rigid-body transformation that perfectly aligns these scans is given by $[\mathrm{R}, \mathbf{t}]$ then the projection of any point in scan **P** onto the reference frame of scan **Q** is given by:

$$\hat{\mathbf{q}}_i = \mathrm{R}\mathbf{p}_i + \mathbf{t}. \tag{4.7}$$

Figure 4.14 Illustration of the nearest neighbor search algorithm used to establish codeword correspondence; each shape above represents a different codeword—green colorings belong to scan **Q** and red to scan **P**. All the codewords in scan **Q** that are within a sphere of radius r around $\hat{\mathbf{q}}_i$ are considered as potential correspondence. The codeword c_i^q that gives the maximum similarity score with c_i^p is chosen as the correspondence.



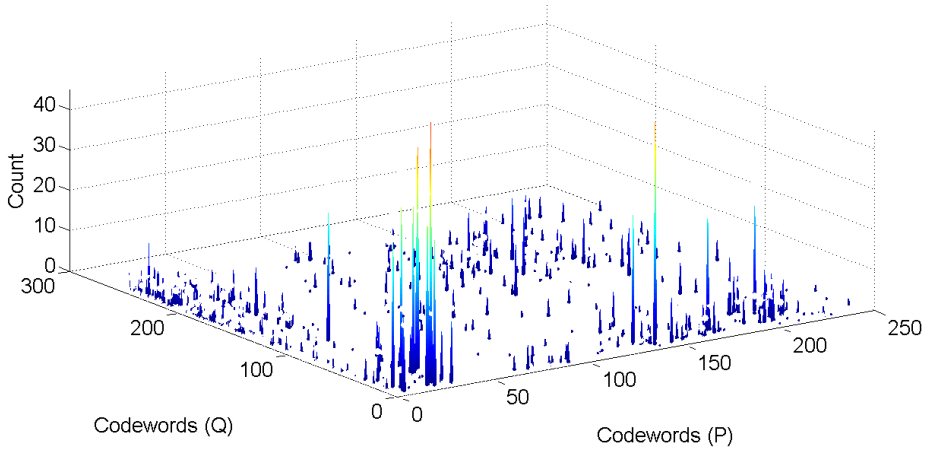
For a correct rigid-body transformation the codeword c_i^p of point $\mathbf{p}_i$ should be the same as the codeword c_i^q of the corresponding point $\hat{\mathbf{q}}_i$. Thus, for a given rigid-body transformation, the corresponding codewords c_i^p and c_i^q are the observations of the random variables X and Y, respectively. We use nearest neighbor search to establish the codeword correspondence (Fig. 4.14). A codeword c_i^p in scan **P** is first projected onto the reference frame of **Q**. All the codewords in scan **Q** that are within a sphere of radius r around c_i^p are considered as potential correspondence. The codeword c_j^q that gives the maximum similarity score with c_i^p is chosen as the correspondence. In case we have multiple codeword assignment within the sphere then the codeword that is closest in Euclidean space to c_i^p takes precedence. We use this correspondence to create the joint histogram of codewords for the given transformation. The MLE of the marginal and joint probabilities of the random variables X and Y can be obtained from the normalized marginal and joint histograms of these codewords.

It is important to note that the number of codewords extracted from a scan (n) are typically much less as compared to the dimensions of the joint histogram ($K \times K$). Moreover, the number of different codewords present in any scan is generally only a fraction of the size of codebook. This causes most of the entries of the joint and marginal histograms to be equal to zero (Fig. 4.15), leading to high mean-squared-error (MSE) in the MLE due to over-fitting. Therefore, we apply a James-Stein (JS) shrinkage approach to improve the MSE of the maximum likelihood (ML) estimator. This method was proposed in [54] for entropy and MI estimation and is based on shrinking the ML estimator of the distribution of a random variable Z toward a target distribution $\mathrm{T} = [\mathrm{T}_1, \mathrm{T}_2, \cdots, \mathrm{T}_K]$:

$$\hat{Z}_k^{JS} = \lambda \mathrm{T}_k + (1 - \lambda)\hat{Z}_k^{ML}, \tag{4.8}$$

where $\hat{Z}_k = p_Z(z = k)$ and $\lambda \in [0, 1]$ is a shrinkage coefficient used to optimize the esti-

Figure 4.15 Sparse joint histogram of codewords. In this example, the size of the joint histogram is $256 \times 256 = 65,536$ and the number of codewords used to populate this histogram are less than 2000.



mation of MI. The target distribution here refers to the distribution of codewords observed in an ideal case (i.e., when $n \gg K \times K$). If all the features extracted from a scan were equally likely then a uniform distribution becomes an obvious choice for the target distribution. However, it is not true in this case, because the occurrence of any feature extracted from the scans is dependent upon the environment. So, we learn the target distribution from the training dataset along with the codebook. The target distribution is estimated from the normalized histogram of codewords present in the training dataset. A sample target distribution corresponding to a particular codebook is shown in Fig. 4.16.

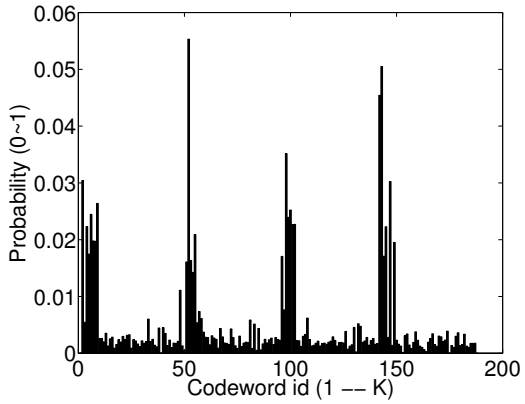
Once we have a good estimate of the joint and marginal probability distributions we can write the MI of the random variables (X, Y) as a function of the rigid-body transformation $[\mathrm{R}, \mathbf{t}]$, thereby formulating a cost function:

$$\mathbf{\Theta} = \arg\max_{\mathbf{\Theta}} \mathrm{MI}(X, Y; \mathbf{\Theta}), \tag{4.9}$$

where $\mathbf{\Theta} = [x, y, z, \phi, \theta, \psi]^\top$ is the six degree of freedom (DOF) parametrization of the rigid-body transformation $[\mathrm{R}, \mathbf{t}]$. Here MI between X and Y is computed in terms of the entropies of these random variables as described in Chapter II Section 2.3.

It should be noted that the JS estimate is a weighted average of two very different estimators. The ML estimate ($\hat{Z}_k^{ML}$) has low bias but since it is estimated from small samples it has a large variance. Whereas the target distribution (T_k) is more biased and is less variable. Therefore, the weight or the shrinkage coefficient (λ) is chosen in a data-driven manner such that $\hat{Z}_k^{JS}$ has small MSE with respect to both $\hat{Z}_k^{ML}$ and T_k. The optimal

Figure 4.16 Marginal target distribution estimated from the training dataset. The feature descriptors chosen here is a combination of FPFH and SURF extracted from the co-registered lidar and camera data. The size of the codebook is 200. Here we intend to show that the target distribution is not uniform, depicting that there are certain features which are observed more frequently in the given environment (urban).



shrinkage coefficient calculated by minimizing this MSE is given by [143, 124]:

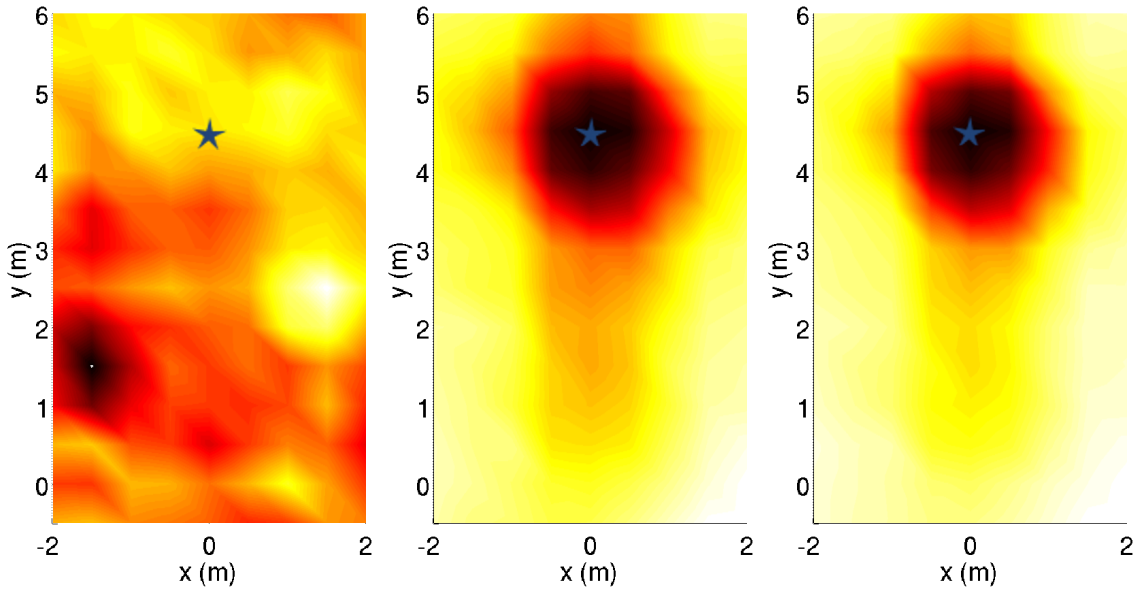
$$\lambda_s = \frac{\mathrm{Var}(\hat{Z}^{ML}) - \mathrm{Cov}(\hat{Z}^{ML}, \mathrm{T})}{\sum_{k=1}^{K} \mathrm{E}[(\hat{Z}_k^{ML} - \mathrm{T}_k)^2]}. \tag{4.10}$$

The optimal shrinkage coefficient is obtained by calculating the first and second order moments of the target distribution (T) and the ML estimate ($\hat{Z}^{ML}$), which has a computational complexity $O(K^2)$. In order to reduce this computation time we propose a shrinkage coefficient λ_e that is directly calculated from the number of codewords obtained from nearest neighbor search in $O(1)$ time:

$$\lambda_e = \frac{2}{(1 + \exp^{-c/\sigma})} - 1, \tag{4.11}$$

where c is the number of corresponding codewords and σ is a parameter proportional to the average number of codewords present in a scan. σ is empirically estimated based on the features used (e.g., if we use SIFT features then typically there are 500–1500 usable SIFT features per scan, so the value of σ is set to 1000). Thus, λ_e takes on a value between 0 (no correspondence / no shrinkage) and 1 (maximum correspondence / full shrinkage). In Fig. 4.17 we show that the MI-based cost function has similar convexity and smoothness when calculated from the JS estimate with shrinkage coefficient either λ_s or λ_e. Since the computation time required to calculate λ_e ($O(1)$) is very small as compared to λ_s ($O(K^2)$), we will use the proposed shrinkage coefficient (λ_e) unless specified otherwise.

Figure 4.17 Top view of the MI cost-function surface versus the translation parameters x and y aligning the two scans. The correct value of translation is given by $(0.02, 4.31)$. Light to dark represents increasing values of the cost function. (a) MI is calculated from the ML estimate of the probability distribution. (b) MI is caluculated from the JS estimate of the probability distribution with standard shrinkage coefficient (λ_s) estimated by minimizing the MSE is used here. (c) MI is calculated from the JS estimate of the probability distribution with proposed shrinkage coefficient (λ_e). The proposed shrinkage coefficient shows similar results as the standard one and requires smaller computation time.



(a) ML estimator (b) JS estimator with λ_s (c) JS estimator with λ_e

The small number of codewords present in a scan make the estimation of MI a challenging task. The shrinkage approach described above provides a robust estimate of MI. Clearly, the proposed shrinkage optimized MI-based cost function shows a global maxima at the desired rigid-body transformation (Fig. 4.17(c)). We use the simplex method proposed by Nelder and Mead [115] to estimate the optimum value of the registration parameter $\mathbf{\Theta}$ that maximizes the cost function given in (4.9). The complete algorithm is summarized in Algorithm 4.

4.4 Experiments and Results

We present results from real data collected from a 3D laser scanner (Velodyne HDL-64E) and an omnidirectional camera system (Point Grey Ladybug3) mounted on the roof of a Ford F-250 vehicle. We use the pose information available from a high-end IMU (Applanix POS-LV 420 INS with Trimble GPS) as the ground-truth to compare the scan

Algorithm 4 Automatic registration of scans by maximization of Mutual Information (MI)

Input: Co-registered camera and lidar scans $\mathbf{P}$ and $\mathbf{Q}$. Initial guess of the rigid-body transformation $\mathbf{\Theta}_0$.
Output: Estimated registration parameter $\{\mathbf{\Theta}\}$.
Extract generalized feature vectors from scans $\mathbf{P}$ and $\mathbf{Q}$.
Quantize the feature vectors using the pre-computed *codebook*.
while convergence of Nelder-Mead simplex optimization **do**
    Calculate correspondence of codewords for the current transformation $\mathbf{\Theta}_k$.
    Calculate the marginal and joint histogram of the corresponding codewords.
    Calculate shrinkage coefficient λ (4.11).
    Calculate James-Stein estimator of the marginal and joint distributions (4.8).
    Calculate the MI: $\mathrm{MI}(X, Y; \mathbf{\Theta}_k)$.
    Update $\mathbf{\Theta_k} \rightarrow \mathbf{\Theta_{k+1}}$.
end while

alignment errors. The details about the datasets used in our experiments are provided in Appendix C. The dataset is divided into two distinct runs: *(i) Downtown* and *(ii) Ford Campus*, both taken in Dearborn, Michigan. We use the *Downtown* dataset for testing and the *Ford Campus* dataset for learning the codebook and the target distribution. We performed the following experiments to analyze the robustness of the proposed algorithm.

4.4.1 Effect of using Data from both Modalities (Camera/Lidar)

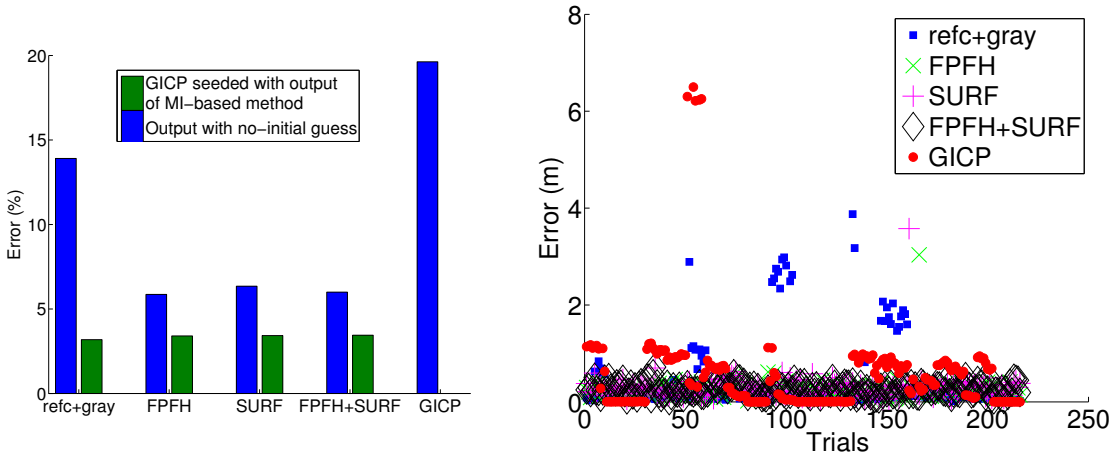
In this experiment we demonstrate the effect of choice of features on the robustness of the algorithm. We show that incorporating features from both modalities (camera/lidar) into the registration process improves the performance. We tested our algorithm for the following features:

1. *Reflectivity and Grayscale (refc+gray):* We used approximately 20,000 uniformly sampled points from the textured scan. The reflectivity obtained from the lidar and the corresponding grayscale intensity obtained from the camera are used as a two-dimensional feature descriptor.

2. *3D only (FPFH):* Keypoints were detected using the Harris (3D) keypoint detection algorithm available in the point cloud library (PCL) [140]. The number of keypoints extracted from a point cloud were between 500–1000.

3. *Image only (SURF):* We used OpenCV's implementation of SURF to extract image keypoints. We assigned the corresponding SURF descriptor to all 3D points that projected within 1-pixel of these keypoints. Only a fraction of the 3D points were assigned these SURF features ($\sim$500–1000).

Figure 4.18 Translational error in MI-based scan alignment algorithm. (a) The blue bars depict the mean registration error starting from no initial guess. The error is calculated as the percentage of the distance between the scans that are aligned (i.e., $error = \frac{\|\mathbf{t}-\hat{\mathbf{t}}\|}{\|\mathbf{t}\|} \times 100$, where $\mathbf{t}$ = true translation vector; $\hat{\mathbf{t}}$ = estimated translation vector; $\|\cdot\|$ = Euclidean norm). The green bars represent the mean error for the same set of scans aligned using GICP seeded with the output obtained from the proposed MI-based algorithm. The GICP algorithm alone does not converge in the absence of a good initial guess (far right error bar). (b) Here we have plotted the translation error ($\|\mathbf{t} - \hat{\mathbf{t}}\|$) for each trial. The proposed MI-based algorithm works well in all trials when we use high-dimensional features. In the case of simple features (refc+gray), the algorithm often gets trapped in a local minima similar to the GICP algorithm (see red circles and blue squares).

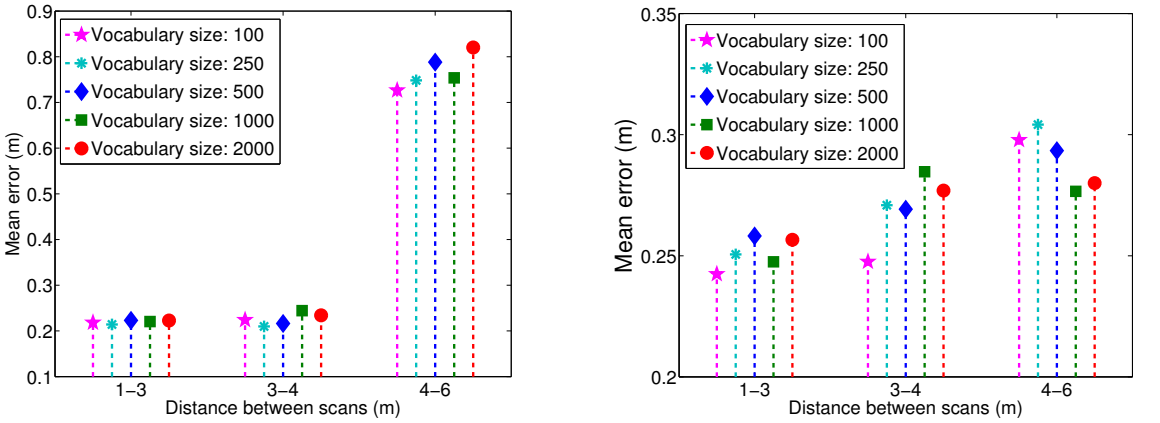


(a) Mean translational error (b) Translational error for each trial

4. *3D and Image combined (FPFH+SURF):* For all the 3D points that are associated to a SURF descriptor, we calculate the FPFH and append it to the existing SURF descriptor.

In this experiment we randomly selected 200 scan-pairs from the *Downtown* dataset spaced approximately 1–5 m apart. We aligned these scan-pairs using the proposed algorithm without any initial guess (i.e., initial guess was fixed at $[0, 0, 0, 0, 0, 0]^\top$). In Fig. 4.18 we have plotted the translation error in the output of the proposed algorithm for different kinds of features used. We also compared the output of the proposed algorithm with the GICP algorithm that uses the 3D point cloud alone. We found that for a poor initial guess, the GICP algorithm fails to converge whereas the proposed algorithm gives better convergence. As shown in Fig. 4.18(a) the average error is reduced when we use high-dimensional features instead of simple surface reflectivity values. If we look at the error in each trial (Fig. 4.18(b)), then we see that the algorithm converges (close to the optimum) in all trials when high-dimensional features are used. However, for simple features (refc+gray), the

Figure 4.19 Mean error in translation for MI-based scan alignment is plotted as a function of distance between the scans for different vocabulary sizes of two different features: (a) refc+gray and (b) FPFH+SURF.



(a) Features used: refc+gray

(b) Features used: FPFH+SURF

algorithm is often trapped in a local minima similar to the GICP algorithm (see red circle and blue squares in Fig. 4.18(b)). The average error in the proposed algorithm can be further reduced by passing its output as an initial guess to the GICP algorithm (the green bars in Fig. 4.18(a)). Thus, the proposed method provides a principled way to incorporate any kind of features into the registration process that helps in reducing the registration error.

4.4.2 Effect of vocabulary size

In this experiment we analyze the effect of vocabulary size (i.e., the quantization levels of the codebook) on the proposed algorithm. Since we are not trying to do any recognition, we do not need very fine quantization (i.e., large codebook size), and can use a coarse codebook. Moreover, the computation time of our algorithm increases with the size of the codebook. Therefore, we would like to keep the size of the codebook as small as possible.

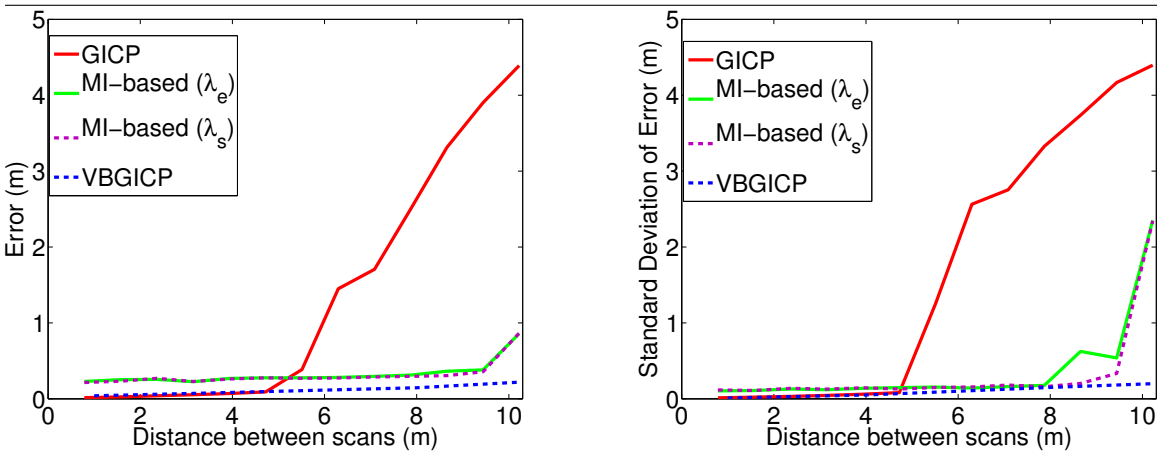
With this experiment we try to identify the optimum size of the codebook for a particular choice of features. We learned the codebook of different sizes $(100, 250, \cdots, 1000)$ for each particular feature set (e.g., refc+gray, FPFH+SURF). We randomly selected 150 scan-pairs (1–3 m, 3–4 m and 4–6 m apart) from the *Downtown* dataset. We aligned these scan-pairs using the proposed algorithm (no initial guess) with different codebooks to quantize the features. In Fig. 4.19 we have plotted the mean translation error for the different codebook sizes. As shown in Fig. 4.19 the average error increases with the distance between the scans (although, for simple features (refc+gray), plotted on top panel of Fig. 4.19, the increase in error is much more than high dimensional features). The effect of vocabulary size as seen in Fig. 4.19 is dependent upon both features used to create the codebook as well as

the distance between the scans under consideration. For example, we found that the optimum value of codebook size for the combined 3D and image features (i.e., FPFH+SURF) is 100 when the distance between the scans is less than 4m, but for larger distances between the scans a finer codebook (vocabulary size = 1000) gives better results. Since the computation complexity of our algorithm is directly proportional to the codebook size, we use smaller codebook sizes as the gain in accuracy is not very large.

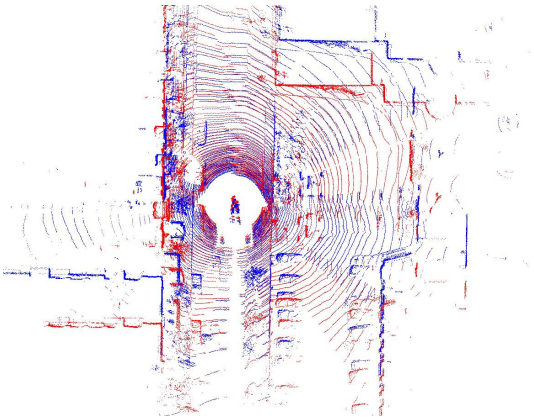
4.4.3 Comparison with GICP and VB-GICP

In this experiment we show that the proposed MI-based cost function has a wider basin of convergence as opposed to the state-of-the-art GICP algorithm [146]. Here, we selected a series of 15 consecutive scans from the *Downtown* dataset. The average distance between the consecutive scans is approximately $0.5\ \mathrm{m}$–$1.0\ \mathrm{m}$. In this experiment we fixed the first scan to be the reference scan and then tried to align the remaining scans (2–15) with the base scan using *(i)* GICP, *(ii)* VB-GICP and *(iii)* MI-based method (with λ_e and λ_s). The average error in translational motion between the base scan and the remaining scans obtained from these algorithms is plotted in Fig. 4.20, computed over 90 trials. We found the plotted error trend to be typical across all of our experiments—in general the GICP algorithm alone would fail after approximately $4\ \mathrm{m}$ of displacement when not fed an initial guess. The reason for this becomes more clear by analyzing the cost function of the MI-based and GICP algorithm. In Fig. 4.21 we have plotted the cost function of the MI-based algorithm and the GICP algorithm for two scans that are $4.5\ \mathrm{m}$ apart. Clearly, the MI-based method has a wider basin of attraction in both x and y direction. The GICP-based cost function (plotted in Fig. 4.21(b)) has a narrow basin of attraction in the y direction but shows better convergence along the x direction. This is mainly due to the nature of the GICP cost, which allows sliding along planar surfaces. The ground plane and the planar structures on both sides of the road (Fig. 4.21(d)) does not constrain the translation along the y direction but it constrains the motion in x direction. Unlike GICP cost, the proposed method does not suffers from planar structures and provides a wider basin of attraction in all directions, thereby converging to the correct solution even if the initial guess is extremely poor. Whereas the GICP cost function shows better convexity near the global maxima, it has a poor basin of convergence. This means the GICP algorithm will converge faster if the initial guess is close to the global maxima but will fail to converge otherwise. This is also the reason for good performance of the VB-GICP algorithm, the RANSAC step calculates a good initial guess (within the narrow convergence basin) for the GICP algorithm which converges to the correct solution.

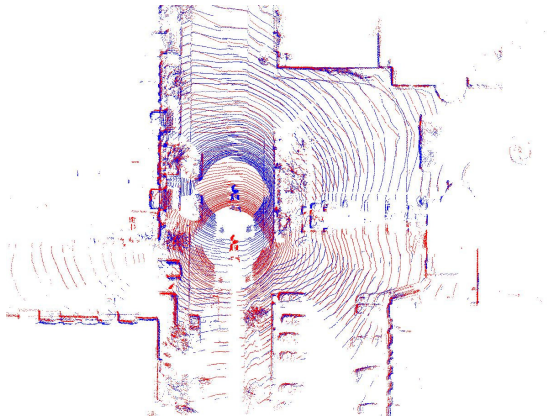
Figure 4.20 Error comparison between GICP, VB-GICP and proposed MI-based method with (FPFH+SURF) features. (a) Graph showing the error and standard deviation in translation as the distance between scans **P** and **Q** is increased. (b)–(c) Top view of the 3D scans aligned with the output of GICP and proposed method for two scans that are approximately 6 m apart. Note that the GICP algorithm fails to align the two scans after approximately 4 m whereas the proposed method shows better convergence property and aligns scans that are almost 10 m apart.



(a) Error comparison between GICP, VB-GICP and proposed MI-based method



(b) Registration result for GICP

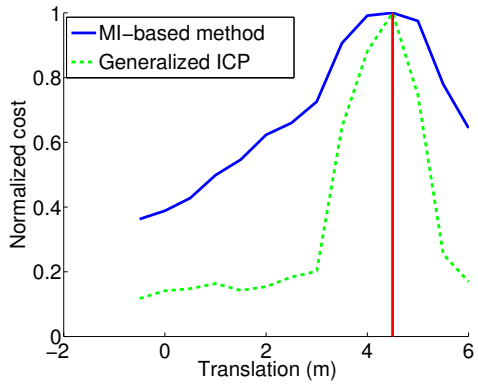


(c) Registration result for MI

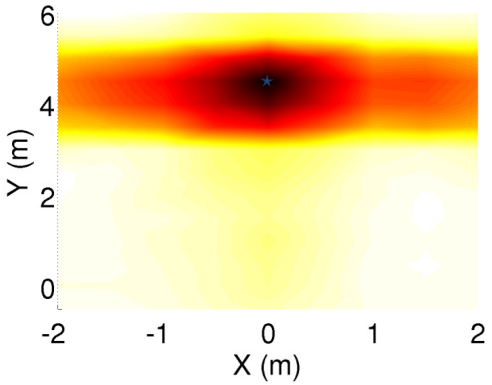
4.5 Conclusion

In this chapter we discussed the problem of scan registration and ways to incorporate visual information from camera imagery into the registration process. We utilized the targetless sensor calibration technique described in the previous chapter to project 3D points from lidar onto the corresponding image (and vice versa). This allowed us to associate high-dimensional feature descriptors from the image (SIFT, SURF, etc.) to the corresponding 3D lidar point that projects onto that pixel location. The 3D points augmented with the

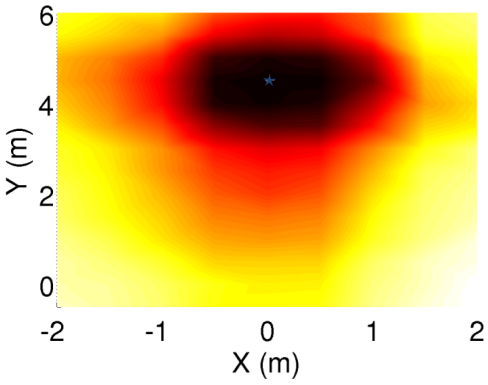
Figure 4.21 In (a) we have plotted cost as a function of translation along y direction (i.e., the direction of motion of vehicle). In (b) and (c) we have plotted the cost function for the same scans by varying both x and y parameters of the rigid-body transformation, while keeping the remaining parameters to be fixed to the true value. The proposed MI-based method (c) has a wider basin of attraction in both x and y direction, whereas the GICP-based cost function (b) has a narrow basin of attraction in the y direction. The basin of attraction in the x direction (b) is better in this case mainly due to the vertical buildings present on both sides (d).



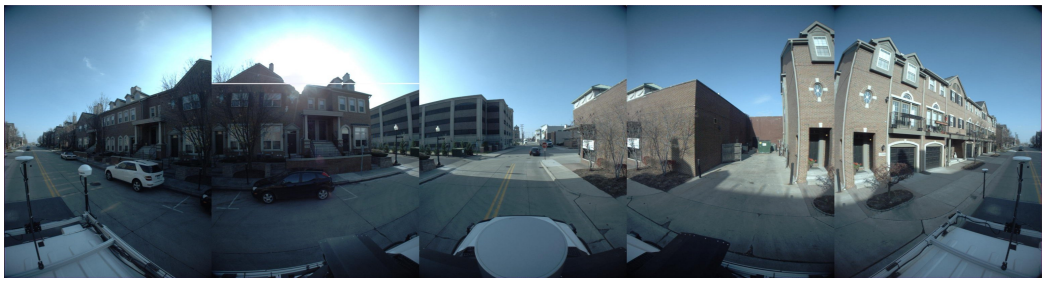
(a) Cost function comparison 1D



(b) Generalized ICP



(c) Proposed MI-based method



(d) Image corresponding to the scans for which the cost functions are evaluated above.

high-dimensional feature vectors are then used to align the two scans.

Here, we presented two methods to estimate the rigid-body transformation that aligns two scans. The first method used a two-step bootstrapping strategy based on image features. In the first step, putative correspondences were established in the high-dimensional feature space and then these correspondences were used within a RANSAC framework to obtain an initial alignment of the two scans. In the second step, this coarse alignment was refined using a generalized ICP [146] framework. Although this algorithm works well and out-performs the state-of-the-art generalized ICP algorithm in situations where the initial guess is not available, it uses the data from the two modalities in a very decoupled manner. Moreover, it requires special assistance to establish putative correspondence in the form of the camera correspondence matrix derived from the omni-directional camera geometry. The second method, on the other hand, does not make any assumptions on the camera geometry and uses a novel MI-based framework for in-corporating complementary information obtained from camera and lidar modalities into the registration process.

Experimental results from real data suggests that both methods have good convergence properties and a wider capture basin than the generalized ICP algorithm. However, we believe that the MI-based method provides a more principled way of fusing lidar and camera modality. In the MI-based method the fused data is directly incorporated into the cost function and since it relies on the statistics of the extracted features, we can further enhance this algorithm by using tools from probability and information theory.

CHAPTER V

Robust Place Recognition

5.1 Introduction

In the previous chapter we described registration of sequentially captured (co-registered) lidar and camera data. Registration of sequential scans provides an estimate of the motion of the vehicle, which can be used to create highly accurate 3D maps of the environment. These 3D maps are used by robots to automatically navigate through that environment by registering current sensor data with previously perceived data in the prior map. Today, robots are required to operate in an environment for days, months or even years. One important task that any robot needs to perform in order to navigate through these environments is to recognize places it has visited before. This place recognition capability has a wide range of applications in autonomous navigation including global localization and loop-closure detection for simultaneous localization and mapping (SLAM) [5, 16, 21, 61]. The task of place recognition in a dynamic environment becomes extremely challenging as a single location appears different over time. The drastic changes in environmental appearance due to changing seasons (summer, fall, winter, etc.), lighting conditions, and dynamical objects make the task of place recognition very challenging (Fig. 5.1).

Most place recognition literature in the mobile robotics community has focused on obtaining correct loop-closures for SLAM. In these situations, the robot creates a map of an *a priori* unknown environment while simultaneously localizing itself in this map. Therefore, the robot has to recognize a place that has been recently visited or added to the map. The time difference between the two instances is usually small and hence the change in appearance of the environment is not too large (apart from change in viewpoint). Vision-based algorithms based on Bag-of-Words techniques [152, 120] have been successfully used for robust place recognition in scenarios like this. Cummins and Newman [31] presented a probabilistic framework, Fast Appearance-Based Mapping (FAB-MAP), that is robust to perceptual aliasing for appearance-based place recognition over maps as big as $1000\ \mathrm{km}$

Figure 5.1 Sample imagery extracted from three different datasets captured in December 2009, October 2010 and February 2011; each row corresponds to the same place. The datasets exhibit significant visual changes due to different weather conditions, lighting and dynamical objects.

long. Pronobis et al. [134] described a fully supervised method for place recognition that is robust to different illumination conditions in indoor scenes. Sunderhauf and Protzel [159] proposed a simple appearance-based place recognition system based on Binary Robust Independent Elementary Feature (BRIEF) descriptors and showed that its performance is comparable to FAB-MAP for large-scale SLAM problems.

Recently, the problem of long-term navigation in a changing environment has received significant attention in the mobile robotics community. The ability to recognize places across seasons, with significant appearance changes (e.g., Fig. 5.1) is very important for long-term autonomy. Glover et al. [40] presented a combination of FAB-MAP [31] and the biologically inspired RatSLAM [104] approach, and showed that it is robust to illumination and structural changes in outdoor environments. Milford and Wyeth [105] proposed to match sequences of images instead of a single image and showed good precision in recognizing places across different seasons (e.g., summer-rain). Churchill and Newman [27] introduced the concept of plastic maps (i.e., a composite representation constructed from multiple overlapping experiences). As a robot repeatedly travels through the same environment under different conditions, it accumulates distinct visual experiences that represent the scene variation. They showed good results on a road vehicle operating over a three month period at different times of day, in different weather, and different lighting conditions. Neubert et al. [117] proposed a novel idea of appearance change prediction. They learn the change in the visual appearance of the environment over time and then use this learned knowledge to predict the appearance of any place under different environmental conditions.

The methods mentioned so far are purely vision-based and use camera as the primary sensing modality. However, robots are often equipped with various perception sensors besides camera like lidar, radar, etc. Although these sensors provide useful complementary information to the camera data, they are mostly used independently for place recognition. There have been some attempts to increase the robustness of place recognition in SLAM systems by fusing the multi-modal data at the landmark level [23]. Paul and Newman developed a more robust FAB-MAP 3D algorithm [129] for large-scale SLAM systems by extending the appearance-only FAB-MAP algorithm to incorporate spatial information of the visual features obtained from laser scanners.

Most of the aforementioned methods either use the image data alone or use the data from the two modalities (camera/lidar) in a decoupled way, without exploiting the statistical dependence of the multi-modal data. Here, we present a novel Mutual Information (MI)-based algorithm for automatic place recognition using co-registered 3D lidar and camera imagery obtained from the method described in Chapter III. Our method provides a robust

framework for incorporating complementary information obtained from these modalities into the recognition process.

The remainder of this chapter proceeds as follows: In Section 5.2 we describe the proposed method of automatic place recognition. In Section 5.3 we present results showing the robustness of the proposed method and present a comparison against a standard Bag-of-Words approach. Finally, in Section 5.4 we summarize our findings.

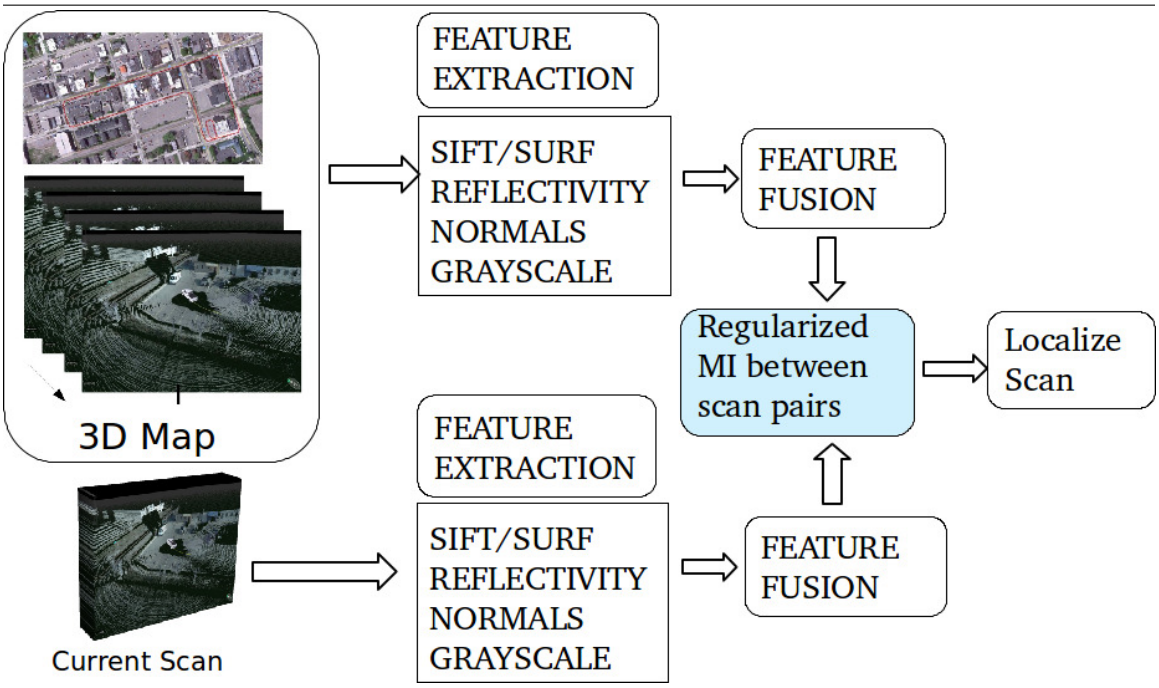
5.2 Methodology

We used data from a 3D laser scanner and a camera system mounted on a mobile robotic platform (see Appendix C) specifically designed for long-term autonomous navigation in a dynamic environment. The robot travels through the environment at different times of the day, in different seasons (summer, fall, winter) and captures time-synchronized lidar and camera data. We assume that the intrinsic and extrinsic calibration parameters for these sensors are either known or estimated beforehand. The extrinsic calibration parameters for the lidar and camera are estimated using the MI-based method described in Chapter III. The calibration of sensors allows us to project 3D points from lidar onto the corresponding camera image (and vice versa). This co-registration allows us to associate features extracted from the camera image (e.g., grayscale value, scale invariant feature transform (SIFT) [96], speeded up robust features (SURF) [13], etc.) to the corresponding 3D lidar point that projects onto that pixel location. The features extracted from the 3D point cloud (e.g., reflectivity, normals, etc.) and camera image are fused together (discussed later in section 5.2.1), and every scan is represented as a collection of these features. Thus, for any two scans corresponding to the same physical location, the joint distribution of these features should show maximum correlation. Here, we use MI as a measure of this correlation along with a simple thresholding scheme to localize the scans within a prior map. An overview of the proposed method is given in Fig. 5.2.

5.2.1 Sensor Data Fusion

In Chapter IV we used a very naive method of sensor data fusion by simply stacking all the features (3D/2D) into a single high-dimensional feature vector. This method worked well in the previous chapter because there we were only registering sequential scans and the little variations in the aligned scans due to dynamic objects was easily handled by the MI-based framework. However, here we want to register sensor data that was captured several days or seasons apart containing significant variations due to change in weather, lighting conditions, dynamic objects and structural changes (e.g., construction). Therefore,

Figure 5.2 Overview of the proposed MI-based robust place recognition algorithm.

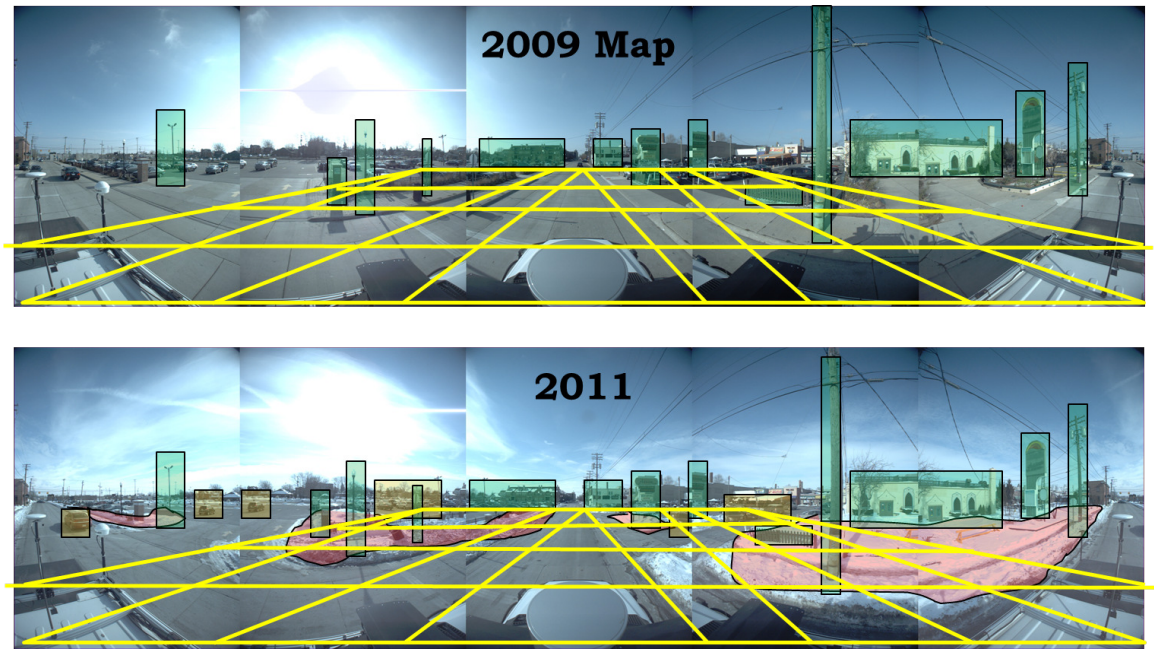


in this section we describe two novel techniques to fuse the features extracted from the co-registered lidar/camera data. We extract both simple features (reflectivity, grayscale, etc.) and high dimensional features (SIFT/SURF) from this data. It is important to note that simple features, like the reflectivity of the 3D points obtained from the lidar, or the intensity of the pixel obtained from the camera, are discrete signals generated by sampling the same physical scene but in a different manner. Since the underlying structure generating these signals is the same, they are statistically dependent upon each other and can be fused together at the *signal level*; however the high-dimensional features (such as SIFT/SURF) from imagery are generally independent from the reflectivity of the lidar point and are therefore fused at the *information level*.

5.2.1.1 Sensor Data Fusion at the Signal Level

In this section we describe a novel method of fusing lidar/camera data at the signal level. The reflectivity from lidar and grayscale intensity from the camera are measurements generated by the same underlying physical scene. These two modalities are therefore highly correlated (i.e., a highly reflective point in lidar data will typically have a high grayscale value for the corresponding pixel). In order to fuse such highly correlated features we divide the 3D scan into voxels (Fig. 5.3) of fixed dimension and calculate the joint-statistics of these simple features extracted from lidar/camera data in each voxel in the form of a

Figure 5.3 The top panel shows the omnidirectional image of a location captured in fall 2009. The bottom panel shows the omnidirectional image of the same location in winter 2011. The significant change in the scene is clearly visible from the two images, for example, snow on the ground (marked in red), dynamic objects (marked in orange), lighting conditions, etc. Such drastic changes make registration of the 2009 and 2011 datasets a challenging problem. However, there are also common objects (marked in green) that have stationary statistics and can be used for registration of sensor data. The 3D space around the sensor is divided into voxels of equal size. Here, we have illustrated the voxelization process in the image via the 2D yellow grid (for visual clarity), however, actual voxels are 3-dimensional. The voxelization allows us to use stationary statistics within each voxel for registration.

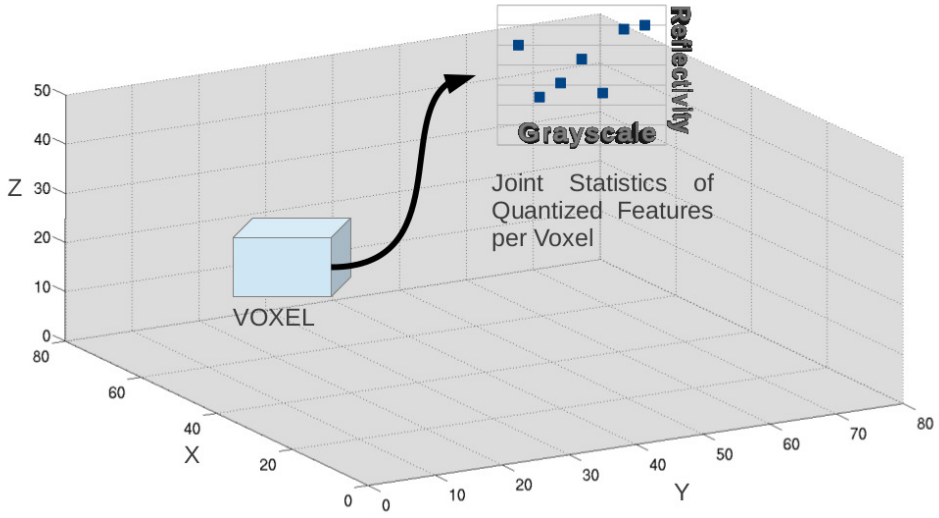


multi-dimensional histogram (Fig. 5.4). This multi-dimensional histogram represents the marginal distribution of the fused features present in the scan, which is later used for estimation of MI. Voxelization of the scene allows us to use stationary statistics within each voxel for robust registration. When we consider the statistics of features across two scans captured at two different times (e.g., fall 2009 and winter 2011), the local stationary statistics per voxel show higher correlation as compared to the global correlation of the entire scene.

5.2.1.2 Sensor Data Fusion at the Information Level

In the previous section we described a method of fusing sensor data that exhibit some correlation (e.g., reflectivity from lidar and grayscale intensity from camera). However,

Figure 5.4 Sensor data fusion at the signal level. The joint statistics of the quantized features present in each voxel constitute the marginal distribution of the features present in the scan.



there are also high-dimensional features (e.g., SIFT/SURF) extracted from the camera data that do not necessarily show any correlation with the reflectivity from the lidar data. They are statistically independent of each other and hence cannot be fused at the signal level; however, they contain useful information necessary for place recognition. Therefore, we propose to fuse these features at the information level by simply computing the total MI between any two scans as the sum of mutual information of each of these independent features,

$$\mathrm{TMI}(P, Q) = \sum_i \mathrm{MI}(\mathcal{F}_i^P, \mathcal{F}_i^Q), \tag{5.1}$$

where $\mathrm{TMI}(P, Q)$ is the total MI between the scans **P** and **Q**, and $\mathcal{F}_i^P$ and $\mathcal{F}_i^Q$ are various features (fused or independent) extracted from the scan data.

5.2.2 Mapping and Place Recognition

We first create a map of the environment from the sensor data. The map consists of equally-spaced scans with known location in a global reference frame. Each scan in the map is a collection of quantized features extracted from the sensor data. Simple features like the reflectivity from lidar and the grayscale intensity values from camera data are integer values, and therefore easy to quantize between a given range (generally $[0 - 255]$ for 8-bit sensors). However, for high-dimensional features (SIFT, SURF, etc.) we first create a dictionary of *codewords* representing the quantization of these features extracted from the scans. We extract N such features (training samples) from a set of scans called the

Figure 5.5 The codebook is learned from the training dataset, and all experiments are performed on the testing dataset. It should be noted that the training and testing datasets are captured in similar outdoor urban environments, though not the same. It is important for the codebook to be representative, but the testing and training environments need not be identical.

(a) Sample images from the training dataset (*Ford Campus*)

(b) Sample images from the testing dataset (*Downtown*)

training dataset (Fig. 5.5). We use a hierarchical k-means clustering [120] algorithm on the training samples to cluster the feature space into K clusters. The centroids of these clusters are defined as *codewords* $\{c_i; i = 1, 2, \cdots, K\}$ and the collection of these codewords is called the *codebook*. We use this codebook to map any feature vector to a unique integer i corresponding to the codeword c_i that gives a maximum similarity score with the feature vector.

We consider the collection of these codewords present in a scan (extracted from the map) as the random variable X. In a given map we have N such scans representing a unique place in the map. The goal of place recognition is to identify the correct location of the robot when it revisits a place in an *a priori* map. Here, we assume that the map is created once and the robot revisits some place in the map after a significant amount of elapsed time. We consider the collection of codewords extracted from this scan (which we will refer to as the query scan) as the random variable Y. The marginal and joint probabilities of these random variables, $p_X(x)$, $p_Y(y)$ and $p_{XY}(x, y)$, can be obtained from the normalized marginal and joint histograms of the codewords present in the scans. Let **Q** be the query scan and **P** be

one of the scans in the map. Let $C^P = \{c_i^p; i = 1, \cdots, n\}$ and $C^Q = \{c_i^q; i = 1, \cdots, m\}$ be the set of codewords, and $\{\mathbf{p}_i; i = 1, \cdots, n\}$ and $\{\mathbf{q}_i; i = 1, \cdots, m\}$ be the set of 3D points corresponding to the codewords present in scans **P** and **Q**, respectively. If the rigid-body transformation that perfectly aligns these scans is given by $[\mathrm{R}, \mathbf{t}]$, then the coordinate transformation of any point in scan **P** onto the reference frame of scan **Q** is given by:

$$\hat{\mathbf{q}}_i = \mathrm{R}\mathbf{p}_i + \mathbf{t}. \tag{5.2}$$

For a correct rigid-body transformation, the codeword c_i^p of point $\mathbf{p}_i$ should be the same as the codeword c_i^q of the corresponding point $\hat{\mathbf{q}}_i$. Thus, for a given rigid-body transformation, the corresponding codewords c_i^p and c_i^q are the observations of the random variables X and Y, respectively.

We use nearest neighbor search method, as described in Section 4.3.1 of Chapter IV, to establish the codeword correspondence. A codeword c_i^p in scan **P** is first transformed to the reference frame of **Q**. All the codewords in scan **Q** that are within a sphere of radius r around c_i^p are considered as potential correspondences. The codeword c_i^q that gives the maximum similarity score with c_i^p is chosen as the correspondence. In the case where we have multiple codeword assignment within the sphere, then the codeword that is closest in Euclidean space to c_i^p takes precedence.

We use the method described above when the 3D location of the codewords is known. However, for certain image features (e.g., SIFT, SURF), their 3D location are often not known due to the sparseness of data obtained from the lidar or due to limited overlap between the field of view of the two sensors. In that case we use the epipolar constraint [53] to establish the correspondence between image features. If $C^P = \{c_i^p; i = 1, \cdots, n\}$ and $C^Q = \{c_i^q; i = 1, \cdots, m\}$ are the set of codewords present in images I^P and I^Q corresponding to scans **P** and **Q**, and $[\mathrm{R}, \mathbf{t}]$ are the rotational and translation parameters between the two cameras, then the two corresponding codewords are related by the epipolar constraint:

$$\tilde{\mathbf{p}}_i^T \mathrm{F} \tilde{\mathbf{q}}_i = 0; \tag{5.3}$$

where $\tilde{\mathbf{p}}_i$ and $\tilde{\mathbf{q}}_i$ are the homogeneous pixel coordinates of the codewords c_i^p and c_i^q, respectively. F is the *fundamental* matrix that maps the codeword in image I^P to the corresponding epipolar line in the image I^Q (Fig. 5.6). Therefore, all the points within certain distance of the epipolar line are considered potential correspondence and the codeword c_i^q that gives the maximum similarity score with c_i^p is taken to be the true correspondence.

We use this correspondence to create the joint histogram of codewords for the given transformation $[\mathrm{R}, \mathbf{t}]$. The maximum likelihood estimate of the marginal and joint prob-

Figure 5.6 The left panel shows a sample codeword (on the lamp-post) extracted from the query image. The right panel shows the epipolar line (green) for the same codeword in the corresponding image from the map. The potential correspondences are marked in blue and the correct correspondence computed based on codeword similarity is marked in red.

(a) (b)

abilities of the random variables X and Y can be obtained from the normalized marginal and joint histograms of these codewords. It is important to note that the number of different codewords present in any scan is generally (especially for high-dimensional features) only a fraction of the size of the codebook. For instance, if we quantize the speeded up robust features features with a vocabulary of size $K = 1,000$, the size of the joint histogram of these codewords will be $[1,000 \times 1,000]$. However, the total number of codewords (n) extracted from a single scan is typically much less than the dimensions of the joint histogram. This causes most of the entries of the joint and marginal histograms to be unobserved, leading to high mean-squared-error (MSE) in the maximum likelihood estimate (MLE) due to over-fitting. In Chapter IV this problem was addressed by using the James-Stein entropy estimator. The target distribution was learned from the training data and the MLE of the distribution was shrinked toward this target to compensate for the bias in the maximum likelihood (ML) estimate of entropy. With the naive sensor data fusion technique used previously, it was easy to learn the target distribution from a training dataset. However, with the sensor data fusion technique used in this chapter, where we create a multi-dimensional spatial histogram of features, learning a target distribution over the entire 3D space is not possible. Since the maximum range of the Velodyne laser scanner is 100 m and the vertical field of view (FOV) of the sensor is $20°$, the dimensions of the viewing cube around the sensor becomes $[200 \text{ m} \times 200 \text{ m} \times 50 \text{ m}]$. If we use voxels of size 1 m and consider the lidar reflectivity and grayscale intensity values quantized between [0, 255], the size of the histogram that needs to be created becomes extremely large

($200 \times 200 \times 50 \times 256 \times 256 = 131,072,000,000$ bins). It is not practical to learn a target distribution for a histogram of such large dimensions from the training data, therefore, we use the Chao-Shen estimator for regularized entropy estimation [24]. This technique has been succesfully used in estimating entropy of gene data in an under-sampled regime with missing species in the observed data. In this approach the entropy of the random variable (with few observations, $n \ll K \times K$) is estimated by applying the Horvitz-Thompson estimator [65] in combination with the Good-Turing correction [125] of the MLE. The Good-Turing-corrected probability estimates are given by:

$$X_k^{GT} = (1 - \frac{m_1}{n}) X_k^{ML}, \tag{5.4}$$

where m_1 is the number of bins with single observation (i.e., $x_k = 1$ and X_k^{ML} is the ML estimate). Combining this with the Horvitz-Thompson estimator, the required entropy is:

$$H^{CS} = -\sum_{k=1}^{n} \frac{X_k^{GT} \log(X_k^{GT})}{(1 - (1 - X_k^{GT})^n)}. \tag{5.5}$$

Once we have a good estimate of the joint and marginal entropies, we can write the total MI of the features present in the two scans as a function of the rigid-body transformation between the scan pair:

$$\mathrm{TMI}(P, Q; \mathbf{\Theta}) = \sum_i \mathrm{MI}(\mathcal{F}_i^P, \mathcal{F}_i^Q; \mathbf{\Theta}), \tag{5.6}$$

where $\mathbf{\Theta} = [x, y, z, \phi, \theta, \psi]^\top$ is the six degree of freedom (DOF) parametrization of the rigid-body transformation $[\mathrm{R}, \mathbf{t}]$. This rigid-body transformation is unknown in the absence of any inertial measurement unit (IMU) or global positioning system (GPS) device. Here we assume that the robot motion is mostly planar, so for every query scan the corresponding scan in the map should be acquired from the same location within a few meters in the x-y plane. Therefore, we perform a linear search over all the scans present in the map dataset with $\mathbf{\Theta} = [0, 0, 0, 0, 0, 0]^\top$ as the transformation parameter. Since we assume planar motion of the vehicle, we also search over certain discrete values of the heading angle (ψ) of the transformation parameters. During this linear search if the TMI is greater than a certain threshold, then we optimize the total MI over the full 6-DOF rigid-body transformation:

$$\hat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\mathrm{argmax}} \sum_j \mathrm{MI}(\mathcal{F}_j^P, \mathcal{F}_j^Q; \mathbf{\Theta}), \tag{5.7}$$

thereby obtaining the exact location of the query scan in the map.

Algorithm 5 Mutual Information based Place Recognition

Input: Co-registered camera and lidar scans, $[\mathbf{P}]_i^N$, constituting the map and query scan, $\mathbf{Q}$.
Output: Scan index from the map that is closest to query scan $\{\mathrm{INDEX}\}$ and its estimated registration parameter, $\{\hat{\mathbf{\Theta}}\}$.
Extract generalized feature vectors from query scan, $\{\mathcal{F}^Q\}$.
Quantize and fuse features.
Let $\mathrm{MAX} = \mathrm{THRESHOLD}$, $\mathrm{INDEX} = 0$;
while i = 1 to N **do**
    Get the quantized feature vectors from map $\{\mathcal{F}^P\} \leftarrow \mathbf{P}_i$.
    for $\psi = 0 : 60^\circ : 360^\circ$ **do**
        $\mathbf{\Theta} \leftarrow [0, 0, 0, 0, 0, \psi]^\top$;
        Calculate the total MI:
        $\mathrm{TMI} = \sum_j \mathrm{MI}(\mathcal{F}_j^P, \mathcal{F}_j^Q; \mathbf{\Theta})$;
        if $\mathrm{TMI} \geq \mathrm{MAX}$ **then**
            $\hat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\operatorname{argmax}} \sum_j \mathrm{MI}(\mathcal{F}_j^P, \mathcal{F}_j^Q; \mathbf{\Theta})$
            $\mathrm{MAX} = \sum_j \mathrm{MI}(\mathcal{F}_j^P, \mathcal{F}_j^Q; \hat{\mathbf{\Theta}})$
            $\mathrm{INDEX} = i$;
        end if
    end for
end while

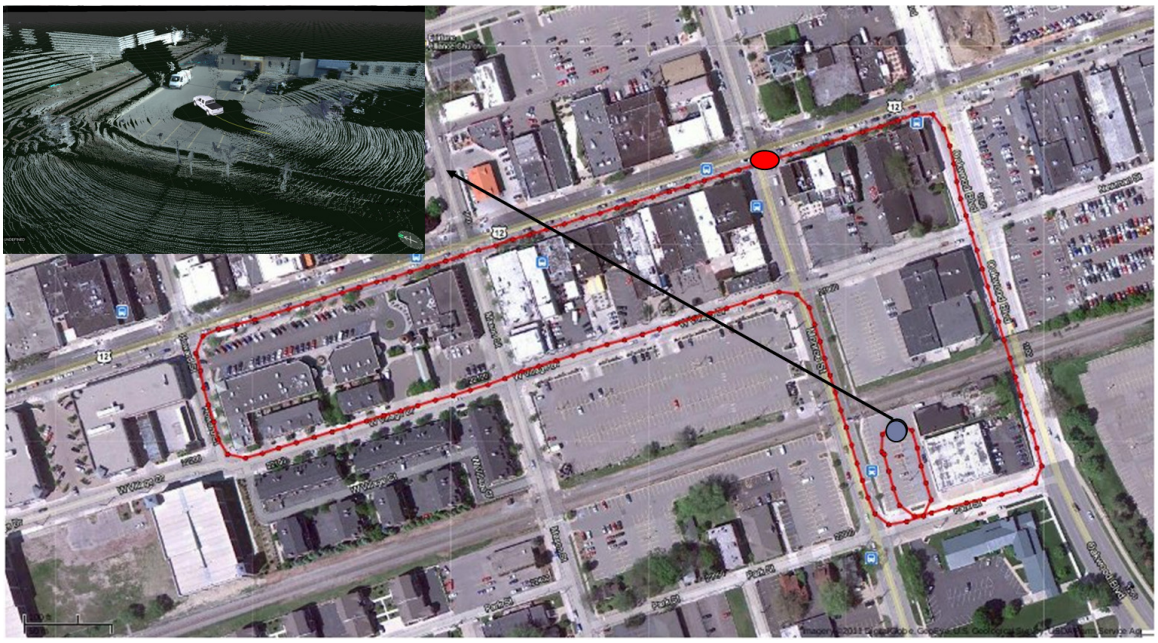
We use the simplex method proposed by Nelder and Mead [115] to estimate the optimum value of the registration parameter, $\mathbf{\Theta}$, that maximizes the cost function given in (5.7). This process is repeated for all the scans in the map and the scan that gives the maximum value of total mutual information with respect to the query scan corresponds to the desired location. The computational complexity of the proposed algorithm depends upon the size of the map that is being searched. Since we do not assume any prior knowledge of the location of the query scan from odometry or any other source, the linear search gets computationally very expensive. The main emphasis of this work though is to show the robustness of a framework that allows to use multi-modal data for recognizing places under significant changes in the appearance of the environment due to changes in weather, lighting, dynamical objects, etc. The complete place recognition method is summarized in Algorithm 5.

5.3 Experiments and Results

We present results from real data collected from a 3D laser scanner (Velodyne HDL-64E) and an omnidirectional camera system (Point Grey Ladybug3) mounted on the roof of a Ford F-250 vehicle (Fig. 5.7). We use the pose information available from a high-end IMU (Applanix POS-LV 420 INS with Trimble GPS) as the ground-truth to compare the

Figure 5.7 The top panel shows the test vehicle (left) mounted with a 3D laser scanner and an omnidirectional camera system (right) as described in Appendix C. The bottom panel shows the 3D map of a section of downtown Dearborn created from the data collected in December 2009. Each node in the map is comprised of a textured 3D point cloud representing a distinct place in the map.

(a) Test vehicle mounted with 3D laser scanner and omnidirectional camera system



(b) 3D map of a section of downtown Dearborn.

place recognition errors. The dataset used in our experiments are divided into two distinct runs: *(i) Downtown* and *(ii) Ford Campus*, both taken in Dearborn, Michigan (details in Appendix C). We have several different sets of data recorded at different times of the year from these locations. In our experiments we have used the *Downtown* dataset for testing and the *Ford Campus* dataset for learning the codebook. We have used five different runs of the *Downtown* dataset recorded in December 2009, September 2010, October 2010,

February 2011 and March 2011 for testing. Each of these runs exhibit significant changes due to weather (e.g., snow on the ground in 2011, no leaves on the trees in December 2009), construction (road blocked, trailers parked) and lighting, thereby making place recognition a challenging task. In our experiments we have used the December 2009 dataset as the prior map (Fig. 5.7(b)) and used scans from the other four datasets as query scans for place recognition. We performed the following experiments to analyze the robustness of the proposed algorithm over a wide variety of input features.

5.3.1 Effect of Using Data from Both Camera and Lidar

In this experiment we demonstrate the effect of feature choice on the robustness of the algorithm. We show that incorporating features from both modalities (camera/lidar) into the registration process improves performance as opposed to individual modalities. We tested our algorithm for the following features:

1. *Reflectivity and Normal:* The reflectivity of the point obtained from lidar and the surface normal at the point are used as features. They are assumed to be independent and fused together at the information level as described in §5.2.1.2.

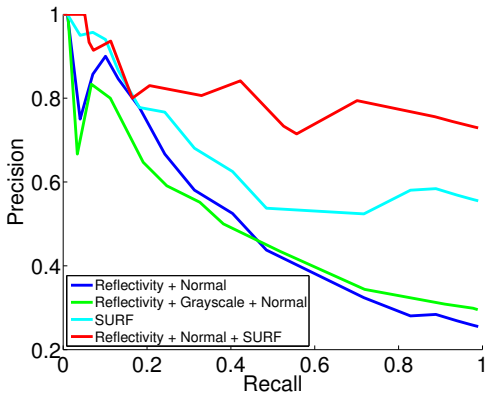
2. *Reflectivity, Grayscale and Normal:* The reflectivity and corresponding grayscale value of a 3D point show high correlation and are fused at the signal level as described in §5.2.1.1. The combined reflectivity and grayscale feature is then fused with the extracted surface normals at the information level (§5.2.1.2).

3. *SURF:* We use OpenCV's [20] implementation of the SURF feature detector and descriptor. It should be noted that we utilize the 3D location of these SURF features to establish correspondences as described in §5.2.2. Therefore, it should not be confused with pure vision-based technique since we are accounting for the 3D location of these features coming from the lidar data.

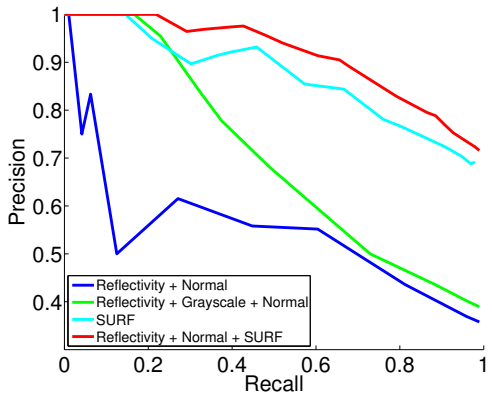
4. *Reflectivity, Normal and SURF:* Here we combine the SURF features with the 3D features (reflectivity and normal). Since SURF features are completely independent of the reflectivity or normal of the 3D point, these features are fused at the information level.

Here we created a prior map from the *Downtown* dataset recorded in December 2009, scans from the data recorded in 2010 and 2011 are treated as query scans. The December 2009 data corresponds to a typical *winter* day with no snow on the ground anywhere and trees without any leaves. We used the scans from the data recorded in 2010 and 2011 as

Figure 5.8 Precision-Recall curves and sample images from the query (2010) and map (2009) datasets.



(a) Precision-Recall curve (Sep. 2010)

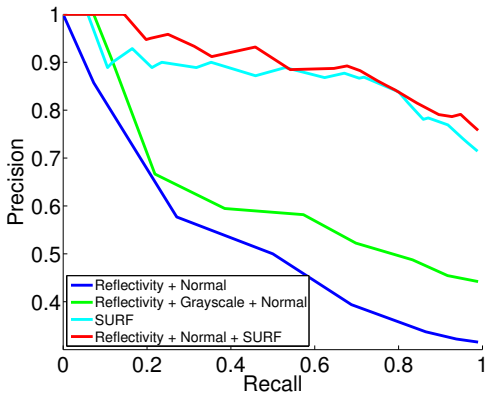


(b) Precision-Recall curve (Oct. 2010)

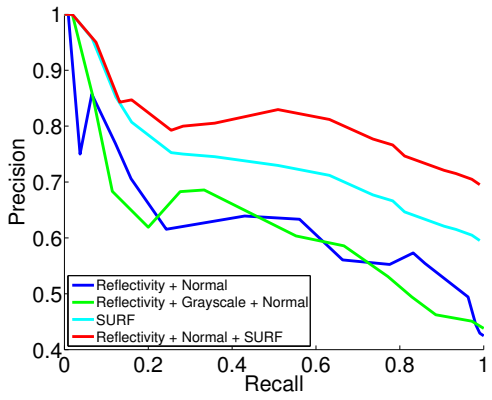
(c) Sample image from query dataset (Sep. 2010)

(d) Corresponding correct image retrieved from map dataset (Dec. 2009)

Figure 5.9 Precision-Recall curves and sample images from the query (2011) and map (2009) datasets.



(a) Precision-Recall curve (Feb. 2011)

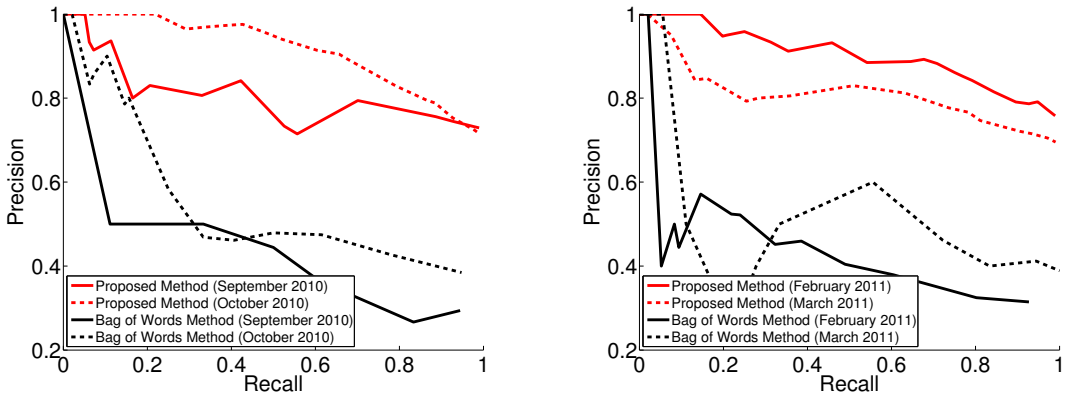


(b) Precision-Recall curve (Mar. 2011)

(c) Sample image from query dataset (Feb. 2011)

(d) Corresponding correct image retrieved from map dataset (Dec. 2009)

Figure 5.10 Comparison with Bag-of-Words method [120].



(a) 2010 dataset. (b) 2011 dataset.

query scans. The query scan is aligned with each scan in the map and the one that gives the highest value of total MI is considered as the best match. In Fig. 5.8(a) and (b), we have plotted the precision-recall curves for the data collected in September and October 2010, respectively. The query dataset here is quite different from the *winter* map dataset not only due to change in dynamical objects (e.g., cars parked on the roads/parking lot), but due to change in weather also. The trees in this dataset are filled with leaves unlike the prior map dataset. Similarly, in Fig. 5.9(a) and (b) we have plotted the precision-recall curves for data collected in February and March 2011, respectively. These datasets collected in winter have snow on the ground and hence exhibit significant change in the appearance of the same location as compared to the map dataset. In both cases we observed that the precision of the proposed algorithm increases as we increase the complexity of features (i.e., using high-dimensional SURF features improves the performance of the algorithm). We also observed an increase in performance when we incorporated data from both modalities. As shown in Fig. 5.8 and Fig. 5.9 lidar data alone (reflectivity and normal) gives a poor precision and recall values, however, if we use the grayscale values in conjunction with lidar reflectivity and normals, we see an improvement in performance. The performance of the algorithm is further improved by using high-dimensional SURF features (instead of grayscale values) that are generally robust to view-point and lighting changes.

5.3.2 Comparison with Bag of Words method

Here we compare the output of the proposed algorithm with the standard bag-of-words algorithm proposed in [120]. We used the same training dataset for learning the vocabulary for both methods. We observe that the proposed algorithm outperforms the bag-of-words

algorithm, which is not surprising since our algorithm takes full advantage of the additional lidar modality. In Fig. 5.10(a) and Fig. 5.10(b) we have plotted the precision-recall curve for 2010 and 2011 datasets, respectively, for both of the methods. The performance of the proposed algorithm (the best output that uses reflectivity, normals and SURF together) is significantly higher as compared to the bag-of-words method. This is mainly because the bag-of-words algorithm only uses the images and does not exploit the 3D information available from the lidar data.

5.4 Conclusion

In this chapter we presented a MI-based place recognition algorithm that allows for the principled fusion of camera and lidar modality information within a single framework. We presented two levels of sensor data fusion *(i)* sensor level and *(ii)* information level. Both of these data fusion techniques take into account the statistical dependence of the features and allows for complete utilization of the information content of the features for robust place recognition. The proposed algorithm showed good results for real data collected from an autonomous vehicle platform, over a period of 3 years at different times of day, under different weather conditions, and with significant lighting and structural changes. The proposed method outperformed the standard image-based technique (bag-of-words) used for place recognition. We showed that using data from multiple modalities can greatly enhance the ability of robot to recognize places within a prior map. The amount of information obtained from one sensor alone is not sufficient for complex tasks like place recognition, especially when the data in the map is significantly different from what the robot perceives in real-time. In situations where the robot operates in changing environments we either need more information, which can be obtained from multi-modal sensors mounted on the robot, or the prior map needs to be updated frequently enough so that the sensor data can be registered with data in the map. It is generally not practical to update maps of large environments so frequently. Therefore, the ability of the proposed algorithm to fuse multi-modal data to recognize places across seasons, with significant appearance changes makes it very suitable for long-term autonomous operation of robots, without the need of updating the prior map.

CHAPTER VI

Conclusions

In this thesis we demonstrated the significance of multi-modal sensors in algorithms required for autonomous navigation of vehicles. We believe that having multiple sensors is necessary for robust autonomous navigation. One type of sensor (e.g., camera, lidar or radar) alone can not provide robust solutions to the problems related to autonomy. Therefore, we need multi-modality sensors that are complimentary in nature. Most of the autonomous vehicle platforms are generally equipped with different modality sensors. However, despite the fact that these sensors provide complimentary information about the surroundings, they are typically used independently. In this thesis we exploit the statistical dependence between the data obtained from different modalities in an information theoretic framework to enhance the robustness of algorithms, including sensor-to-sensor calibration, scan registration and place recognition within a prior map.

6.1 Summary of Contributions

This thesis presents information theoretic solutions to some of the common problems, such as sensor calibration, scan registration and place recognition, encountered in autonomous navigation of vehicles. The work done in this thesis presents a different perspective for multi-modal sensors. We see the sensors as the source of information and utilize the statistical dependence of this information (obtained from different modalities) to solve the following three important problems in autonomy:

6.1.1 Calibration of Sensors to Generate Fused Sensor Data

One of the most important contributions of this thesis is in developing an algorithm for automatic extrinsic calibration of lidar and camera that allows projection of 3D points onto the corresponding camera image. This projection of 3D points onto the image plane

forms the basis of data fusion obtained from these modalities. We presented two different techniques for sensor calibration, one that requires a special target and the other that uses the statistical dependence of the sensor data in an information theoretic framework. The information theoretic algorithm automatically estimates the rigid-body transformation between a camera and 3D laser scanner by maximizing the Mutual Information (MI) between the reflectivity obtained from lidar and grayscale intensity values obtained from a camera image. The most important thing to take away about this algorithm is that it is completely data driven and does not require any artificial targets to be placed in the field-of-view of the sensors.

Generally, sensor calibration in a robotic application is performed once, and the same calibration is assumed to be true for rest of the life of that particular sensor suite. However, for robotics applications where the robot needs to go out into rough terrain, assuming that the sensor calibration is not altered during a task is often not true. Although we should calibrate the sensors before every task, it is typically not practical to do so if it requires to set up a calibration environment every time. The information theoretic algorithm for sensor calibration presented in this thesis is free from any such constraints, and therefore can be easily used to fine tune the calibration of the sensors *in situ*, which makes it applicable to in-field calibration scenarios.

6.1.2 Registration of Sequential Scans Comprised of Fused Sensor Data

Fusion of sensor data allows association of intensity information obtained from camera image with the 3D points obtained from the lidar data. The second contribution of this thesis is in utilizing this fused sensor data for registration of two sequential scans. We presented two methods that align two sequential scans comprised of co-registered camera and lidar data. The first method uses a two-step boot-strapping strategy that utilizes the image features to obtain an initial alignment and then refines this coarse alignment using a pure 3D point based GICP algorithm. Although this method proved to have better convergence as compared to naive generalized ICP (GICP) algorithm, it uses the data from the two modalities in a decoupled way and is also dependent upon the camera geometry to produce good results. The second method, on the other hand, presented a MI-based scan registration algorithm that allows for the principled fusion of camera and lidar modality information within a single optimization framework. The wide spread input flexibility of this algorithm was demonstrated through the use of several different feature sets ranging from very simple (reflectivity + grayscale) to advanced (FPFH+SURF). This algorithm demonstrated good convergence performance and a wider capture basin than state-of-the-art GICP, when implemented with high-dimensional features.

6.1.3 Place Recognition within a 3D Map Comprised of Fused Sensor Data

Registration of sequential scans provides an estimate of the relative motion of the robot that can be used to create rich 3D maps (comprised of fused lidar and camera data) of the environment. The third and final contribution of this thesis is in developing an information theoretic framework for recognizing places within such prior 3D maps. The place recognition algorithm presented here is robust to drastic changes in environmental appearance. The map data was collected once and the vehicle was localized within a previously generated map despite significant changes in the environmental appearance. We have also shown that using data from different modality sensors increase the robustness of recognizing previsited places within an *a priori* map.

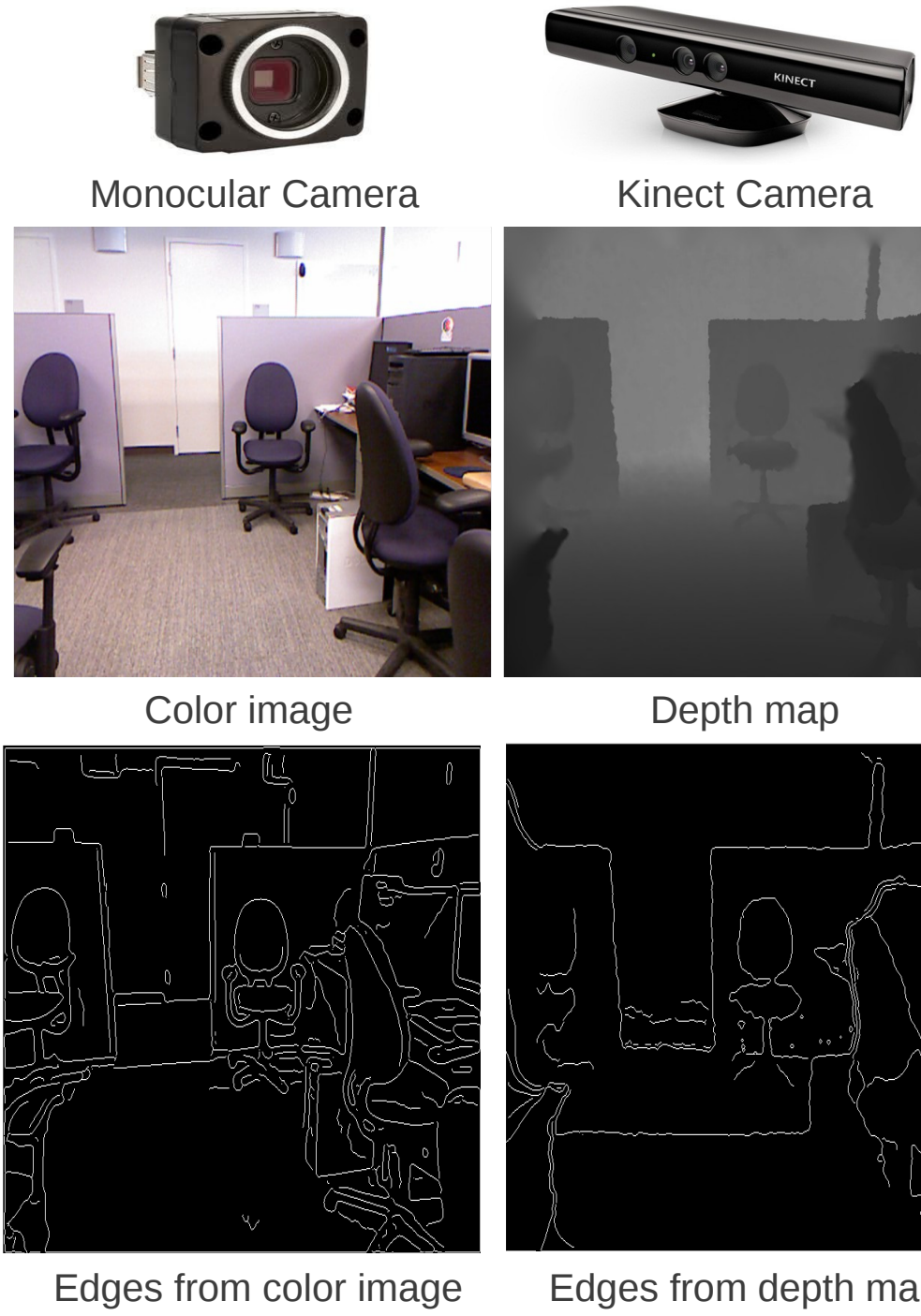
6.2 Future Works

The work presented in this thesis is intended to encourage researchers to use multi-modal sensors and explore the possibilities of utilizing information theoretic concepts to increase the robustness of various robotics algorithms. In this section we discuss some of the immediate logical extensions of the research work presented here.

6.2.1 Extension of MI-based Calibration of Sensors to other Modalities

The MI-based framework for calibration of multi-modal sensors presented in this thesis assumes that the range sensor also provides reflectivity of the surface apart from the range information. However, oftentimes we need to use other sensing modalities (e.g., sonars or laser without reflectivity) due to system constraints or for certain specific requirements. We have not tested the proposed MI-based framework with any sensors that do not provide a direct correlation as observed between reflectivity and grayscale values. However, we believe that one can extract similar features from the two modalities, which can be used in the MI framework. For instance, if the lidar just gives the range returns (i.e., no reflectivity), then we can first generate a depth map from the point cloud. The depth map and the corresponding image should both have edge and corner features at the discontinuities in the environment (Fig. 6.1). The MI between these features should exhibit a maxima at the sought after rigid-body transformation. There might be even better ways to extract highly correlative features for such sensors and it will be worthwhile to explore the use of these features in the MI-based calibration framework.

Figure 6.1 Here we illustrate an example extension of the MI-based calibration framework to a monocular camera with a Kinect camera, which does not provide any reflectivity information. The color image and the corresponding depthmap from the Kinect camera are shown below (center panel). The edges extracted from the color image and the corresponding depthmap (bottom panel) clearly show correlation.



6.2.2 Improvement in Optimization Techniques

All the algorithms developed in this thesis are formulated as an optimization problem that maximizes a cost function to estimate some unknown variables. Therefore, the optimization techniques used to solve the unknown variables directly affects the robustness and computational complexity of the algorithms. We mainly used a simple gradient descent technique to solve the optimization problem in most parts of this thesis. It works well in most of the cases, however, it is sensitive to initialization errors and has the tendency to get trapped in a local optima. Moreover, since the cost function in most of our cases is a non-parametric function of the unknown variables the gradient is computed numerically, thereby making it computationally expensive for real-time applications. Hence, it is worthwhile to explore other optimization techniques that can potentially increase the robustness and efficiency of the overall algorithm.

6.2.3 Exploiting Causality of Temporal Data

The place recognition algorithm presented in Chapter V matches the query scan, comprised of co-registered lidar and camera data, with scans in the map. A similarity score based on the MI of the features extracted from the fused lidar and camera data corresponding to the query scan and the scans in the map is calculated. Maximization of this MI score is used to locate the query scan within the map. Here we consider that each scan constitutes a unique location, and the MI between the features extracted from the query scan and every scan within the map is computed independently. Each scan is considered to be independent despite the fact that the lidar and camera data that we obtain from the sensors is actually comprised of a stream of temporal data with a causal structure. The work presented in this thesis does not utilize the causality of the data streams and mainly treats them as independent snapshots of the environment. However, one can utilize this temporal information to increase the robustness of the place recognition algorithm. Therefore, incorporating the causality of the temporal sequence into the place recognition framework constitutes another direction of future research. The MI between two random variables quantifies the statistical dependence of the random variables, however it does not account for the causality of the system of those random variables. The MI-based framework can be extended to incorporate the causality of the data by using directed information (DI) [100] as a measure of statistical dependence for the sequence of fused lidar and camera data streams.

The DI from the sequence $X^{[N]} = \{^iX\}_{i=1}^N$ of random variables to the sequence $Y^{[N]} = \{^iY\}_{i=1}^N$ is a natural extension of the MI between the random variables exhibiting some

causal structure. A representation of DI in terms of conditional entropies is given as [78]:

$$\mathrm{DI}(X^{[N]} \rightarrow Y^{[N]}) = \mathrm{H}(Y^{[N]}) - \mathrm{H}(X^{[N]}||Y^{[N]}) \tag{6.1}$$

where $\mathrm{H}(X^{[N]}||Y^{[N]})$ is the causally conditional entropy

$$\mathrm{H}(X^{[N]}||Y^{[N]}) = \sum_{i=1}^{N} \mathrm{H}(Y_i|Y_{i-1}, X^i). \tag{6.2}$$

The entropy $\mathrm{H}(X)$ of a random variable X denotes the amount of uncertainty in X. Hence (6.1) shows that $\mathrm{DI}(X^{[N]} \rightarrow Y^{[N]})$ is the cumulative reduction in the uncertainty of sequence $Y^{[N]}$ when it is supplemented by the information about the causal sequence $X^{[N]}$. Therefore, if we represent a location in the map by a sequence of scans (instead of a single scan) then we can use the DI between the query sequence and the sequences in the map for place recognition in a similar framework as presented in Chapter V, which should boost the performance of the place recognition algorithm.

APPENDICES

APPENDIX A

Relationship between Mutual Information (MI) and Entropy

In this appendix we derive the relationship between MI and joint and marginal entropies of two random variables X and Y. The MI between two random variables X and Y is defined as the KL-divergence of the joint distribution $p(X, Y)$ with the product of marginals $p(X)p(Y)$:

$$
\begin{aligned}
\mathrm{MI}(X,Y) &= \sum_i \sum_j p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)p(y_j)} && \text{(A.1)} \\
&\quad [\because p(x,y) = p(x|y)p(y); \text{Bayes rule}] && \text{(A.2)} \\
&= \sum_i \sum_j p(x_i, y_j) \log \frac{p(x_i|y_j)}{p(x_i)} && \text{(A.3)} \\
&= -\sum_i \sum_j p(x_i, y_j) \log p(x_i) + \sum_i \sum_j p(x_i, y_j) \log p(x_i|y_j) && \text{(A.4)} \\
&\quad \left[\because \sum_y p(x,y) = p(x); \text{Marginalization}\right] && \text{(A.5)} \\
&= -\sum_i p(x_i) \log p(x_i) - \left(-\sum_i \sum_j p(x_i, y_j) \log p(x_i|y_j) \right) && \text{(A.6)} \\
&= H(X) - \sum_j p(y_j) \left(-\sum_i p(x_i|y_j) \log p(x_i|y_j) \right) && \text{(A.7)} \\
&= H(X) - \sum_j p(y_j) H(X|Y = y_j) && \text{(A.8)} \\
&= H(X) - H(X|Y). && \text{(A.9)}
\end{aligned}
$$

Now, we can write the conditional entropy of X given Y as:

$$
\begin{aligned}
H(X|Y) &= -\sum_j p(y_j) H(X|Y = y_j) && \text{(A.10)} \\
&= -\sum_j p(y_j) \sum_i p(x_i|y_j) \log p(x_i|y_j) && \text{(A.11)} \\
&= -\sum_i \sum_j p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(y_j)} && \text{(A.12)} \\
&= -\sum_i \sum_j p(x_i, y_j) \log p(x_i, y_j) + \sum_i \sum_j p(x_i, y_j) \log p(y_j) && \text{(A.13)} \\
&= H(X, Y) - \left(-\sum_j p(y_j) \log p(y_j) \right) && \text{(A.14)} \\
&= H(X, Y) - H(Y). && \text{(A.15)}
\end{aligned}
$$

Therefore, the mutual information between two random variables is the amount of reduction in the uncertainty of X when we have some knowledge about Y:

$$\mathrm{MI}(X, Y) = H(X) + H(Y) - H(X, Y). \qquad \text{(A.16)}$$

APPENDIX B

Covariance of Estimated Calibration Parameters

In this appendix, we derive the covariance of the estimated calibration parameters for the checkerboard pattern method as described in Chapter III. The covariance of the measurements can be propagated through any nonlinear optimization function with finite first and second order partial derivatives as described in [52]. Let us consider the nonlinear optimization function given by the reprojection error as defined in Section 3.2.1 of Chapter III:

$$F = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\frac{{}^{h}\mathbf{N}_i}{\|{}^{h}\mathbf{N}_i\|} \cdot ({}^{h}_{\ell}\mathrm{R} {}^{\ell}\mathbf{P}_j \ + \ {}^{h}\mathbf{t}_{h\ell}) - \|{}^{h}\mathbf{N}_i\| \right)^2 . \tag{B.1}$$

Here, ${}^{h}\mathbf{t}_{h\ell} = [t_x, t_y, t_z]^\top$ is a Euclidean 3-vector from h to ℓ as expressed in frame h, and ${}^{h}_{\ell}\mathrm{R}$ is a $[3 \times 3]$ orthonormal rotation matrix parametrized by the Euler angles $[\theta_x, \theta_y, \theta_z]$ (which rotates frame ℓ into frame h). The corresponding covariance of the estimated parameters is given by [52]:

$$\Sigma_\Theta = \left[\frac{\partial^2 F}{\partial \Theta^2}(X, \Theta) \right]^{-1} \frac{\partial^2 F^\top}{\partial X \partial \Theta}(X, \Theta) \Sigma_X \frac{\partial^2 F}{\partial X \partial \Theta}(X, \Theta) \left[\frac{\partial^2 F}{\partial \Theta^2}(X, \Theta) \right]^{-1}, \tag{B.2}$$

where $\Theta = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^\top$, $\frac{\partial^2 F}{\partial \Theta^2}$ is a (6×6) matrix operator given by:

$$\frac{\partial^2 F}{\partial \Theta^2} = \begin{bmatrix} \frac{\partial^2 F}{\partial t_x^2} & \frac{\partial^2 F}{\partial t_x t_y} & \frac{\partial^2 F}{\partial t_x t_z} & \frac{\partial^2 F}{\partial t_x \theta_x} & \frac{\partial^2 F}{\partial t_x \theta_y} & \frac{\partial^2 F}{\partial t_x \theta_z} \\ \frac{\partial^2 F}{\partial t_y t_x} & \frac{\partial^2 F}{\partial t_y^2} & \frac{\partial^2 F}{\partial t_y t_z} & \frac{\partial^2 F}{\partial t_y \theta_x} & \frac{\partial^2 F}{\partial t_y \theta_y} & \frac{\partial^2 F}{\partial t_y \theta_z} \\ \frac{\partial^2 F}{\partial t_z t_x} & \frac{\partial^2 F}{\partial t_z t_y} & \frac{\partial^2 F}{\partial t_z^2} & \frac{\partial^2 F}{\partial t_z \theta_x} & \frac{\partial^2 F}{\partial t_z \theta_y} & \frac{\partial^2 F}{\partial t_z \theta_z} \\ \frac{\partial^2 F}{\partial \theta_x t_x} & \frac{\partial^2 F}{\partial \theta_x t_y} & \frac{\partial^2 F}{\partial \theta_x t_z} & \frac{\partial^2 F}{\partial \theta_x^2} & \frac{\partial^2 F}{\partial \theta_x \theta_y} & \frac{\partial^2 F}{\partial \theta_x \theta_z} \\ \frac{\partial^2 F}{\partial \theta_y t_x} & \frac{\partial^2 F}{\partial \theta_y t_y} & \frac{\partial^2 F}{\partial \theta_y t_z} & \frac{\partial^2 F}{\partial \theta_y \theta_x} & \frac{\partial^2 F}{\partial \theta_y^2} & \frac{\partial^2 F}{\partial \theta_y \theta_z} \\ \frac{\partial^2 F}{\partial \theta_z t_x} & \frac{\partial^2 F}{\partial \theta_z t_y} & \frac{\partial^2 F}{\partial \theta_z t_z} & \frac{\partial^2 F}{\partial \theta_z \theta_x} & \frac{\partial^2 F}{\partial \theta_z \theta_y} & \frac{\partial^2 F}{\partial \theta_z^2} \end{bmatrix}_{(6 \times 6)}, \tag{B.3}$$

$X = [{}^{h}\mathbf{N}_1, \{{}^{\ell}\mathbf{P}\}_1, {}^{h}\mathbf{N}_2, \{{}^{\ell}\mathbf{P}\}_2 \ldots {}^{h}\mathbf{N}_i, \{{}^{\ell}\mathbf{P}\}_i, \ldots]^\top$ is the vector of measurements composed of the normals of the observed planes (${}^{h}\mathbf{N}_i$) and the laser points ($\{{}^{\ell}\mathbf{P}\}_i = \{{}^{\ell}p_1, {}^{\ell}p_2, \cdots\}_i$) lying on these planes, and $\frac{\partial^2 F}{\partial X \partial \Theta}$ is a $(K \times 6)$ matrix operator given by:

$$\frac{\partial^2 F}{\partial X \partial \Theta} = \begin{bmatrix} \frac{\partial^2 F}{\partial N_{x_1} \partial t_x} & \frac{\partial^2 F}{\partial N_{x_1} \partial t_y} & \frac{\partial^2 F}{\partial N_{x_1} \partial t_z} & \frac{\partial^2 F}{\partial N_{x_1} \partial \theta_x} & \frac{\partial^2 F}{\partial N_{x_1} \partial \theta_y} & \frac{\partial^2 F}{\partial N_{x_1} \partial \theta_z} \\ \frac{\partial^2 F}{\partial N_{y_1} \partial t_x} & \frac{\partial^2 F}{\partial N_{y_1} \partial t_y} & \frac{\partial^2 F}{\partial N_{y_1} \partial t_z} & \frac{\partial^2 F}{\partial N_{y_1} \partial \theta_x} & \frac{\partial^2 F}{\partial N_{y_1} \partial \theta_y} & \frac{\partial^2 F}{\partial N_{y_1} \partial \theta_z} \\ \frac{\partial^2 F}{\partial N_{z_1} \partial t_x} & \frac{\partial^2 F}{\partial N_{z_1} \partial t_y} & \frac{\partial^2 F}{\partial N_{z_1} \partial t_z} & \frac{\partial^2 F}{\partial N_{z_1} \partial \theta_x} & \frac{\partial^2 F}{\partial N_{z_1} \partial \theta_y} & \frac{\partial^2 F}{\partial N_{z_1} \partial \theta_z} \\ \frac{\partial^2 F}{\partial P_{x_1}^{\,1} \partial t_x} & \frac{\partial^2 F}{\partial P_{x_1}^{\,1} \partial t_y} & \frac{\partial^2 F}{\partial P_{x_1}^{\,1} \partial t_z} & \frac{\partial^2 F}{\partial P_{x_1}^{\,1} \partial \theta_x} & \frac{\partial^2 F}{\partial P_{x_1}^{\,1} \partial \theta_y} & \frac{\partial^2 F}{\partial P_{x_1}^{\,1} \partial \theta_z} \\ \frac{\partial^2 F}{\partial P_{y_1}^{\,1} \partial t_x} & \frac{\partial^2 F}{\partial P_{y_1}^{\,1} \partial t_y} & \frac{\partial^2 F}{\partial P_{y_1}^{\,1} \partial t_z} & \frac{\partial^2 F}{\partial P_{y_1}^{\,1} \partial \theta_x} & \frac{\partial^2 F}{\partial P_{y_1}^{\,1} \partial \theta_y} & \frac{\partial^2 F}{\partial P_{y_1}^{\,1} \partial \theta_z} \\ \frac{\partial^2 F}{\partial P_{z_1}^{\,1} \partial t_x} & \frac{\partial^2 F}{\partial P_{z_1}^{\,1} \partial t_y} & \frac{\partial^2 F}{\partial P_{z_1}^{\,1} \partial t_z} & \frac{\partial^2 F}{\partial P_{z_1}^{\,1} \partial \theta_x} & \frac{\partial^2 F}{\partial P_{z_1}^{\,1} \partial \theta_y} & \frac{\partial^2 F}{\partial P_{z_1}^{\,1} \partial \theta_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 F}{\partial N_{x_i} \partial t_x} & \frac{\partial^2 F}{\partial N_{x_i} \partial t_y} & \frac{\partial^2 F}{\partial N_{x_i} \partial t_z} & \frac{\partial^2 F}{\partial N_{x_i} \partial \theta_x} & \frac{\partial^2 F}{\partial N_{x_i} \partial \theta_y} & \frac{\partial^2 F}{\partial N_{x_i} \partial \theta_z} \\ \frac{\partial^2 F}{\partial N_{y_i} \partial t_x} & \frac{\partial^2 F}{\partial N_{y_i} \partial t_y} & \frac{\partial^2 F}{\partial N_{y_i} \partial t_z} & \frac{\partial^2 F}{\partial N_{y_i} \partial \theta_x} & \frac{\partial^2 F}{\partial N_{y_i} \partial \theta_y} & \frac{\partial^2 F}{\partial N_{y_i} \partial \theta_z} \\ \frac{\partial^2 F}{\partial N_{z_i} \partial t_x} & \frac{\partial^2 F}{\partial N_{z_i} \partial t_y} & \frac{\partial^2 F}{\partial N_{z_i} \partial t_z} & \frac{\partial^2 F}{\partial N_{z_i} \partial \theta_x} & \frac{\partial^2 F}{\partial N_{z_i} \partial \theta_y} & \frac{\partial^2 F}{\partial N_{z_i} \partial \theta_z} \\ \frac{\partial^2 F}{\partial P_{x_1}^{\,i} \partial t_x} & \frac{\partial^2 F}{\partial P_{x_1}^{\,i} \partial t_y} & \frac{\partial^2 F}{\partial P_{x_1}^{\,i} \partial t_z} & \frac{\partial^2 F}{\partial P_{x_1}^{\,i} \partial \theta_x} & \frac{\partial^2 F}{\partial P_{x_1}^{\,i} \partial \theta_y} & \frac{\partial^2 F}{\partial P_{x_1}^{\,i} \partial \theta_z} \\ \frac{\partial^2 F}{\partial P_{y_1}^{\,i} \partial t_x} & \frac{\partial^2 F}{\partial P_{y_1}^{\,i} \partial t_y} & \frac{\partial^2 F}{\partial P_{y_1}^{\,i} \partial t_z} & \frac{\partial^2 F}{\partial P_{y_1}^{\,i} \partial \theta_x} & \frac{\partial^2 F}{\partial P_{y_1}^{\,i} \partial \theta_y} & \frac{\partial^2 F}{\partial P_{y_1}^{\,i} \partial \theta_z} \\ \frac{\partial^2 F}{\partial P_{z_1}^{\,i} \partial t_x} & \frac{\partial^2 F}{\partial P_{z_1}^{\,i} \partial t_y} & \frac{\partial^2 F}{\partial P_{z_1}^{\,i} \partial t_z} & \frac{\partial^2 F}{\partial P_{z_1}^{\,i} \partial \theta_x} & \frac{\partial^2 F}{\partial P_{z_1}^{\,i} \partial \theta_y} & \frac{\partial^2 F}{\partial P_{z_1}^{\,i} \partial \theta_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{(K \times 6)}, \tag{B.4}$$

where $K = 3 \times (m+n)$, and m is the number of plane normals and n is the total number of 3D points used. If we substitute the values of $\frac{\partial^2 F}{\partial \Theta^2}$ and $\frac{\partial^2 F}{\partial X \partial \Theta}$ into (B.2), we obtain the $[6 \times 6]$ covariance matrix of the estimated parameters. The diagonal elements of this covariance matrix yields the variance in the translational and rotational components of the estimated calibration parameters.

APPENDIX C

Ford Campus Vision and Lidar Dataset

The University of Michigan and Ford Motor Company have been collaborating on autonomous ground vehicle research since the 2007 DARPA Urban Grand Challenge, and through this continued collaboration have developed an autonomous ground vehicle testbed based upon a modified Ford F-250 pickup truck (Fig. C.1). This vehicle was one of the finalists in the 2007 DARPA Urban Grand Challenge and demonstrated the ability to navigate in a mock urban environment, which included moving targets, intermittently blocked pathways, and regions of denied global positioning system (GPS) reception [101]. This allowed us to collect large scale visual and inertial data of some real-world urban environments, which might be useful in generating rich, textured, 3D maps of the environment for navigation purposes.

Here we present two datasets collected by this vehicle while driving in and around the Ford research campus and downtown Dearborn in Michigan. The data includes various small and large loop closure events, ranging from feature-rich downtown areas to feature-poor empty parking lots. The most significant aspect of the data is the precise co-registration of 3D laser data with omnidirectional camera imagery, thereby adding visual information to the structure of the environment as obtained from the laser data. The fused vision data along with odometry information constitutes an appropriate framework for benchmarking various state-of-the-art computer vision and robotics algorithms. We hope that this dataset will be useful to the robotics and vision community and will provide new research opportunities by using the image and laser data together, along with the odometry information.

We have published a paper titled "*Ford campus vision and lidar dataset*" describing this dataset. The paper is published in the International Journal of Robotics Research and the dataset is being used by robotics researchers all over the world. The remainder of this appendix describes this dataset in detail.

Figure C.1 The Ford F-250 pickup truck with perception and inertial sensors strategically mounted on the vehicle for autonomous navigation research.

C.1 Sensors

We used a modified Ford F-250 pickup truck as our base platform. Although, the large size of the vehicle might appear like a hindrance in the urban driving environment, it has proved useful because it allows strategic placement of different sensors. Moreover, the large space at the back of the truck was sufficient to install four 2U quad-core processors along with a ducted cooling mechanism. The vehicle was integrated with the following perception and navigation sensors:

Perception Sensors

- *Velodyne HDL-64E lidar* [165] has two blocks of lasers each consisting of 32 laser diodes aligned vertically, resulting in an effective 26.8° vertical field of view (FOV). The entire unit can spin about its vertical axis at speeds up to 900 rpm (15 Hz) to provide a full 360 degree azimuthal field of view. The maximum range of the sensor is 120 m and it captures about 1 million range points per second. We captured our dataset with the laser spinning at 10 Hz.

- *Point Grey Ladybug3 omnidirectional camera* [82] is a high resolution omnidirectional camera system. It has six 2-Megapixel (1600×1200) cameras with five positioned in a horizontal ring and one positioned vertically. This enables the system to collect video from more than 80% of the full viewing sphere. The camera can capture images at multiple resolutions and multiple frame rates, it also supports hardware JPEG compression on the head. We collected our dataset at half resolution (i.e., 1600×600) and 8 fps in raw format (uncompressed).

- *Riegl LMS-Q120 lidar* [136] has an 80° FOV with very fine (0.2° per step) resolution. Two of these sensors are installed at the front of the vehicle and the range returns and the intensity data corresponding to each range point are recorded as the laser sweeps across the FOV.

Navigation Sensors

- *Applanix POS-LV 420 INS with Trimble GPS* [7] is a professional-grade, compact, fully integrated, turnkey position and orientation system combining a differential GPS, an inertial measurement unit (IMU) rated with 1° of drift per hour, and a 1024-count wheel encoder to measure the relative position, orientation, velocity, angular rate and acceleration estimates of the vehicle. In our dataset we provide the 6-DOF pose estimates obtained by integrating the acceleration and velocity estimates provided by this system at a rate of 100 Hz.

- *Xsens MTi-G* [171] is a consumer-grade, miniature size and low weight 6-DOF micro-electro-mechanical system (MEMS) IMU. The MTi-G contains accelerometers, gyroscopes, magnetometers, an integrated GPS receiver, static pressure sensor and a temperature sensor. Its internal low-power signal processor provides real-time and drift-free 3D orientation as well as calibrated 3D acceleration, 3D rate of turn, and 3D velocity of the vehicle at 100 Hz. It also has an integrated GPS receiver that measures the GPS coordinates of the vehicle. The 6-DOF pose of the vehicle at any instance can be obtained by integrating the 3D velocity and 3D rate of turn.

C.2 Data Capture

In order to minimize the latency in data capture, all of the sensor load is evenly distributed across the four quad-core processors installed at the back of the truck. Time synchronization across the computer cluster is achieved by using a simple network time

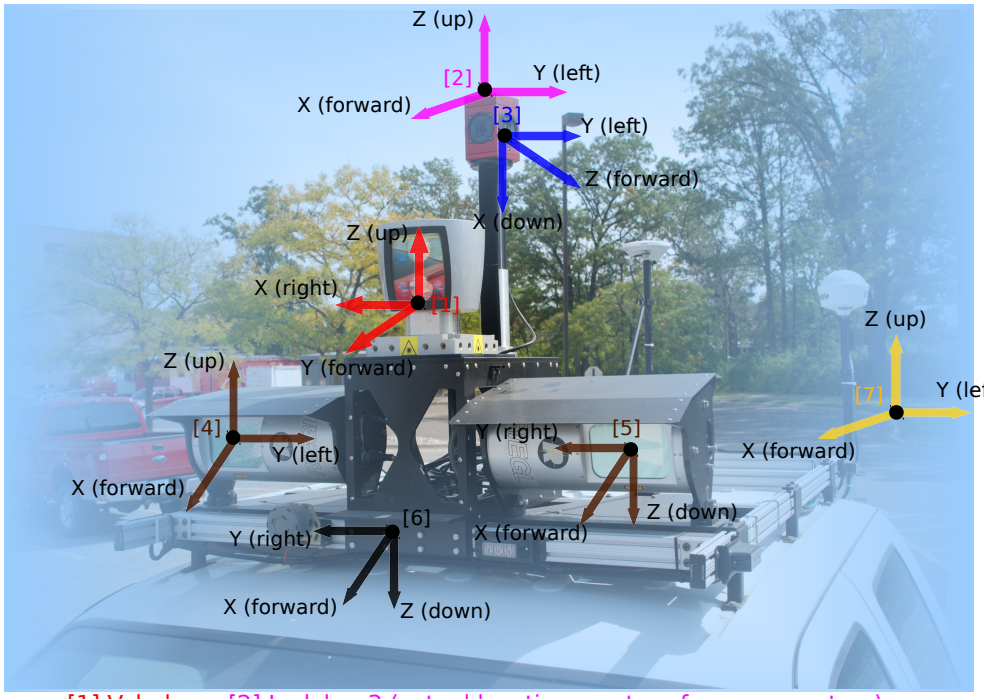
protocol (NTP) [107] like method whereby one computer is designated as a "master" and every other computer continually estimates and slews its clock relative to the master's clock. This is done by periodically exchanging a small packet with the master and measuring the time it takes to complete the round trip. We estimate the clock skew to be upper bounded by 100 µs based upon observed round-trip packet times. When sensor data is received by any of these synchronized host computers, it is timestamped along with the timestamp associated with the native hardware clock of the sensor device. These two timestamps are then merged according to the algorithm documented in [122] to determine a more accurate timestamp estimate. This sensor data is then packaged into a lightweight communication and marshalling (LCM) [66] packet and is transmitted over the network using multicast user datagram protocol (UDP). This transmitted data is captured by a logging process, which listens for such packets from all the sensor drivers, and stores them on the disk in a single log file. The data recorded in this log file is timestamped again, which allows for synchronous playback of the data later on.

C.3 Sensor Calibration

All of the sensors (perceptual and navigation) are fixed to the vehicle and are related to each other by static coordinate transforms. These rigid-body transformations, which allow the re-projection of any point from one coordinate frame to the other, were calculated for each sensor. The coordinate frames of the two navigation sensors (Applanix and MTi-G) coincide and are called the body frame of the vehicle—all other coordinate frames are defined with respect to the body frame (Fig. C.2). Here we use the Smith, Self and Cheeseman [153] coordinate frame notation to represent the 6-DOF pose of a sensor coordinate frame where $X_{ab} = [x, y, z, roll, pitch, yaw]^\top$ denotes the 6-DOF pose of frame b with respect to frame a. The calibration procedure and the relative transformation between the different coordinate frames are described below and summarized in Table C.1. We also use the concept of a local frame [113], which is a smoothly varying coordinate system with arbitrary origin. The vehicle moves in this local frame according to the best available relative motion estimate coming from the IMU.

- **Relative transformation between the Velodyne laser scanner and body frame (X_{bl}):** A research grade coordinate-measuring machine (CMM) was used to precisely obtain the position of some known reference points on the truck with respect to the body frame, which is defined to be at the center of the rear axle of the truck. The measured CMM points are denoted X_{bp}. Typical precision of a CMM is of the order of micrometers, thus for all practical purposes we assumed that the relative position (X_{bp}) of these reference

Figure C.2 Relative position of the sensors with respect to the body frame.



[1] Velodyne, [2] Ladybug3 (actual location: center of camera system),
[3] Ladybug3 Camera 5, [4] Right Riegl, [5] Left Riegl,
[6] Body Frame (actual location: center of rear axle)
[7] Local Frame (Angle between the X-axis and East is known)

points obtained from CMM are true values without any error. We then manually measured the position of the Velodyne from one of these reference points to get X_{pl}. Since the transformation X_{pl} is obtained manually, the uncertainty in this transformation is of the order of a few centimeters, which for all practical purposes can be considered to be 2–5 cm. The relative transformation of the Velodyne with respect to the body frame is thus obtained by compounding the two transformations [153].

$$X_{bl} = X_{bp} \oplus X_{pl} \tag{C.1}$$

- **Relative transformation between the Riegl Lidars and the body frame (Left Riegl = X_{bR_l}, Right Riegl = X_{bR_r}):** These transformations are also obtained manually with the help of the CMM as described above in C.3.

- **Relative transformation between the Velodyne laser scanner and Ladybug3 camera head (X_{hl}):** This transformation allows us to project any 3D point in the laser reference frame into the camera head's frame and thereby into the corresponding camera image.

Table C.1 Relative transformation of sensors

Transform	Value (meters and degrees)
X_{bl}	$[2.4, -0.01, -2.3, 180°, 0°, 90°]$
X_{bR_l}	$[2.617, -0.451, -2.2, 0°, 12°, 1.5°]$
X_{bR_r}	$[2.645, 0.426, -2.2, 180°, 6°, 0.5°]$
X_{hl}	$[0.3, -0.005, -0.426, -0.15°, 0.00°, -90.27°]$
X_{bh}	$[2.06, 0.0, -2.72, -180°, -0.02°, -0.8°]$

These parameters can be estimated by either the target-based method or the MI-based targetless method described in Chapter III. The transformation obtained using the MI-based method is given in Table C.1.

- **Relative transformation between the Ladybug3 camera head and body frame (X_{bh}):** Once we have X_{bl} and X_{hl}, the transformation X_{bh} can be calculated using the compounding operation [153]:

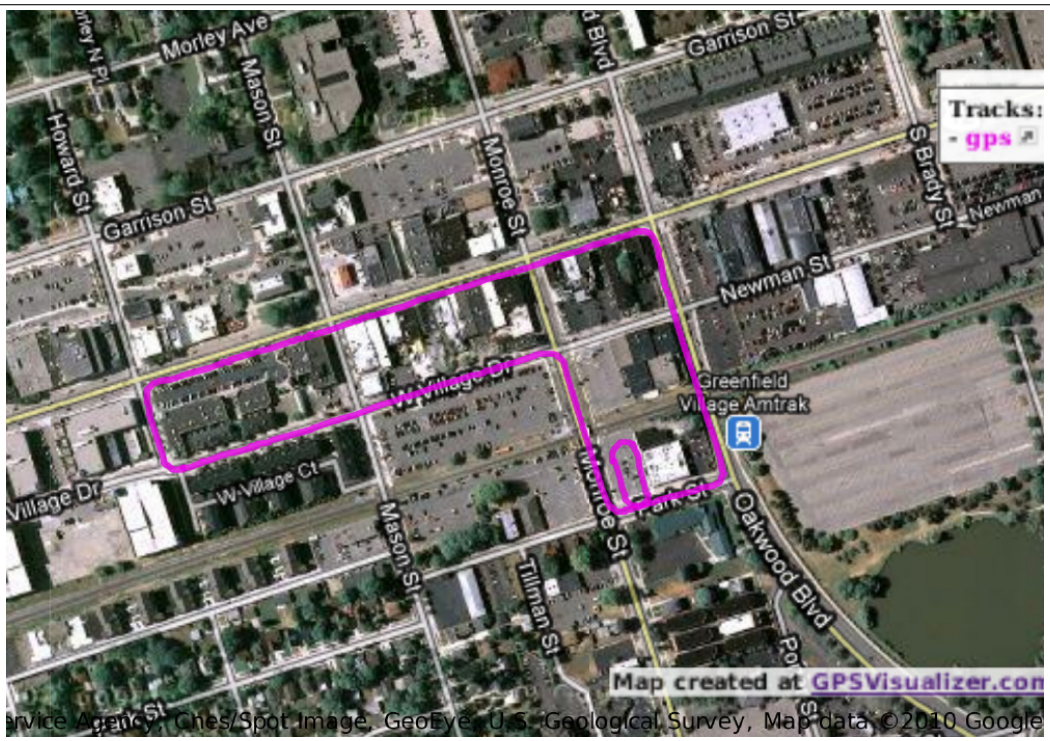
$$X_{bh} = X_{bl} \oplus (\ominus X_{hl}). \tag{C.2}$$

C.4 Data Collection

The data was collected around the Ford Research Campus area and downtown area in Dearborn, Michigan, henceforth referred to as the test environment. It is an ideal dataset representing an urban environment and it is our hope that it will be a useful dataset to researchers working on autonomous perception and navigation in unstructured urban scenes.

The data was collected while driving the modified Ford F-250 around the test environment several times while covering different areas. We call each data collection exercise to be a trial. In every trial we collected the data keeping in mind the requirements of state of the art simultaneous localization and mapping (SLAM) algorithms, thereby covering several small and large loops in our dataset. A sample trajectory of the vehicle in one of the trials around downtown Dearborn and around Ford Research Complex is shown in Fig. C.3. The unprocessed data consists of three main files for each trial. One file contains the raw images of the omnidirectional camera, captured at 8 fps, and the other file contains the timestamps of each image measured in microseconds since 00:00:00 Jan 1, 1970 Coordinate Universal Time (UTC). The third file contains the data coming from remaining sensors (perceptual/navigational). This file stores the data in a LCM log file format similar to that described in [66]. The unprocessed data consist of raw spherically distorted images from the omnidirectional camera and the raw point cloud from the lidar (without any compensation for the vehicle motion). Here we present a set of processed data with MATLAB

Figure C.3 The top panel shows the trajectory of the vehicle in one trial around downtown Dearborn. The bottom panel shows the trajectory of the vehicle in one trial around Ford Research Complex in Dearborn. Here we have plotted the GPS data coming from the Trimble overlaid atop of an aerial image from Google maps.



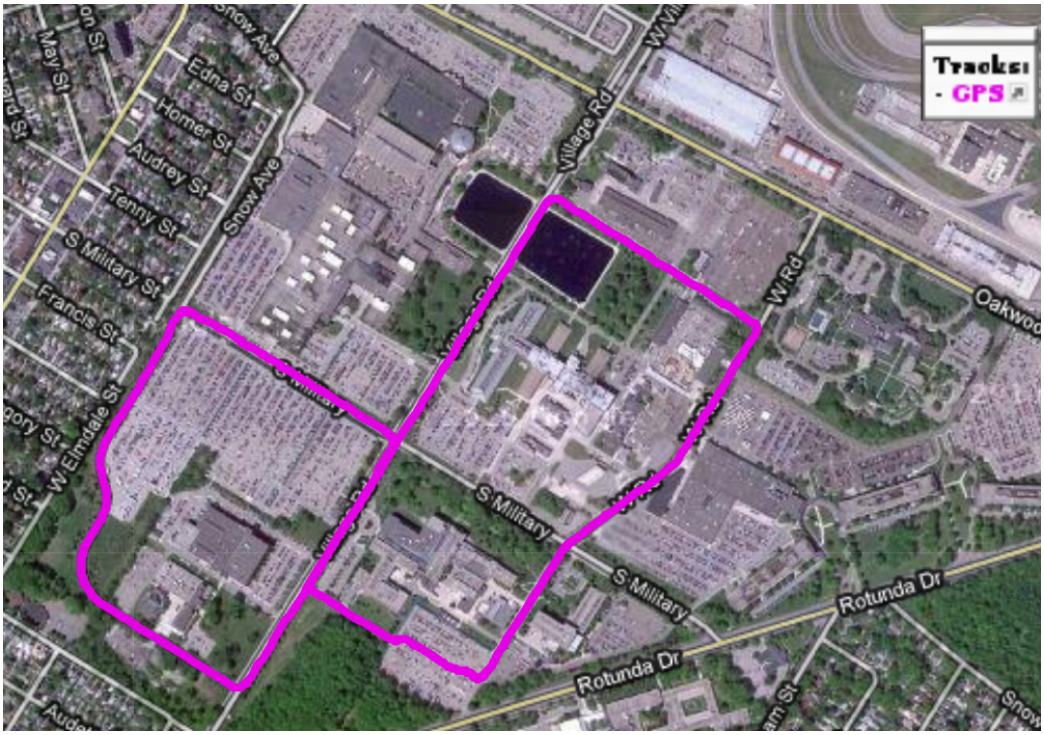
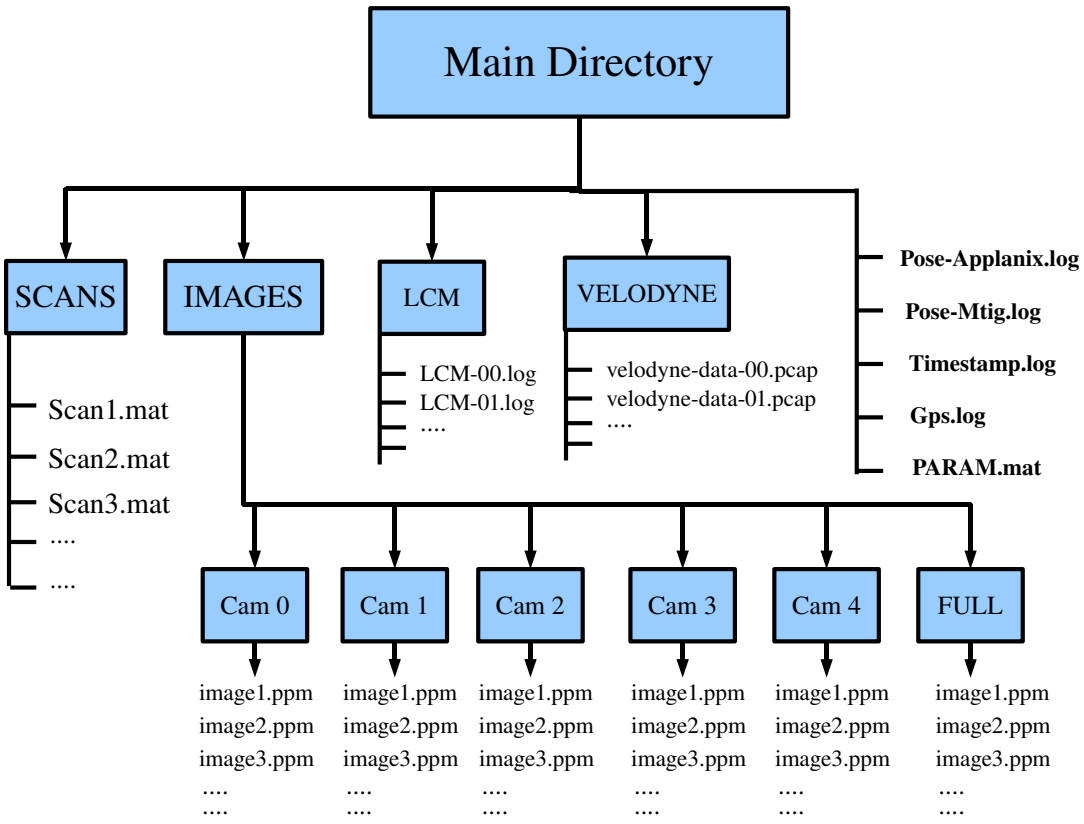


Figure C.4 The directory structure containing the dataset. The rectangular blocks represent folders.



scripts that allow easy access of the dataset to the users. The processed data is organized in folders and the directory structure is as shown in Fig. C.4. The main files and folders are described below:

- **LCM**: This folder contains the LCM log file corresponding to each trial. Each LCM log file contains the raw 3D point cloud from the Velodyne laser scanner, the lidar data from the two Reigl LMS-Q120s and the navigational data from the navigational sensors described in Section C.1. We provide software to playback this log file and visualize the data in an interactive graphical user interface.

- **Timestamp.log**: This file contains the Unix timestamp, measured in microseconds since 00:00:00 Jan 1, 1970 UTC, of each image captured by the omnidirectional camera during one trial.

- **Pose-Applanix.log**: This file contains the 6-DOF pose of the vehicle in a local coordinate frame, as described in [113]. The local frame is arbitrarily fixed at the location where

the Applanix is initialized for the first time (Ford parking lot) and all the vehicle poses are reported with respect to this local frame. The angle θ that the X-axis of this local frame makes with East is known and is recorded in the Gps.log (see Gps.log below). The local frame is thus an East-North-Up (ENU) coordinate frame, but rotated by an angle θ. The acceleration and rotation rates given by the IMU (Applanix POS-LV) are first transformed into the local frame and then integrated to obtain the pose of the vehicle in this local reference frame. The Pose-Applanix.log when loaded in MATLAB has the following fields for each vehicle pose:

- *Pose.utime*: is the Unix timestamp measured in microseconds.
- *Pose.pos*: is the 3-DOF position of the vehicle in the local reference frame.
- *Pose.rph*: is the roll, pitch and heading of the vehicle. Here the heading is with respect to the orientation of the vehicle when the Applanix is initialized. It is not the true heading calculated with respect to East. In order to get the true heading you need to subtract the angle that the local frame makes with East, which is given in the GPS.log.
- *Pose.vel*: is the 3-DOF velocity of the vehicle in the local reference frame.
- *Pose.rotation_rate*: is the angular velocity of the vehicle.
- *Pose.accel*: is the 3-DOF acceleration of the vehicle in local reference frame.
- *Pose.orientation*: is the orientation of the vehicle given in quaternions.

We have not fused the IMU data with GPS in our dataset, but we provide GPS data separately along with the uncertainties in GPS coordinates. We have not provided the uncertainty in the pose estimates in our dataset but it can be calculated from the measurement noise obtained from the Applanix POS-LV specification sheet [7].

- **Pose-Mtig.log**: This file contains the navigational data: 3D rotational angles (roll, pitch and yaw), 3D accelerations and 3D velocities of the vehicle along with the timestamp provided by the Xsens MTi-G, during one trial. Here we provide the vehicle pose estimated by integrating the velocities. The raw data is available in the LCM log file and can be extracted from there. The Pose-Mtig.log when loaded in MATLAB has the following fields for each vehicle pose:
  - *Pose.utime*: is the Unix timestamp measured in microseconds.
  - *Pose.pos*: is the 3-DOF position of the vehicle in North-East-Down (NED) coordinate frame with origin at the position where the vehicle starts.

Figure C.5 The distorted images, from the five horizontal sensors of the omnidirectional camera, stacked together.

- *Pose.rph*: is the Euler roll, pitch and heading of the vehicle. Heading here is reported with respect to North.
- *Pose.vel*: is the 3-DOF velocity of the vehicle in the NED reference frame.
- *Pose.rotation_rate*: is the angular velocity of the vehicle.
- *Pose.accel*: is the 3-DOF acceleration of the vehicle.

- **Gps.log**: This file contains the GPS data of the vehicle along with the uncertainties provided by the Trimble GPS, obtained during one trial. This data structure has the following fields:
  - *Gps.utime*: is the Unix timestamp measured in microseconds.
  - *Gps.lat_lon_el_theta*: is the [4x1] array of GPS coordinates. The first three entries are the latitude, longitude and elevation/altitude whereas the last entry *theta* is the angle that the X-axis of the local frame makes with East (i.e., θ).
  - *Gps.cov*: is the [4x4] covariance matrix representing the uncertainty in the GPS coordinates.

- **IMAGES**: This folder contains the undistorted images captured from the omnidirectional camera system during one trial. The folder is further divided into sub-folders containing images corresponding to individual cameras from the omnidirectional camera system. This folder also contains a folder named "FULL", which contains the distorted images stacked together in one file as depicted in Fig. C.5.

- **PARAM.mat**: This contains the intrinsic and extrinsic parameters of the omnidirectional camera system and is a (1×5) array of structures with the following fields:
  - *PARAM.K*: This is the (3×3) matrix of the internal parameters of the camera.
  - *PARAM.R, PARAM.t*: These are the (3×3) rotation matrix and (3×1) translation vector, respectively, which transforms the 3D point cloud from the laser reference system to the camera reference system. These values were obtained by compounding the following transformations:

$$X_{c_i l} = X_{c_i h} \oplus X_{hl}, \tag{C.3}$$

    where $X_{c_i h}$ defines the relative transformation between camera head and the i^{th} sensor (camera) of the omnidirectional camera system. The transformation $X_{c_i h}$ is precisely known and is provided by the manufacturers of the camera.
  - *PARAM.MappingMatrix*: This contains the mapping between the distorted and undistorted image pixels. This mapping is provided by the manufacturers of the Ladybug3 camera system and it corrects for the spherical distortion in the image. A pair of distorted and undistorted images from the camera is shown in Fig. C.6.

- **SCANS**: This folder contains 3D scans from the Velodyne laser scanner, motion compensated by the vehicle pose provided by the Applanix POS-LV 420 IMU. Each scan file in this folder is a MATLAB file (.mat) that can be easily loaded into the MATLAB workspace. The structure of individual scans once loaded in MATLAB is shown below:
  - *Scan.XYZ*: is a $(3 \times N)$ array of the motion compensated 3D point cloud represented in the Velodyne's reference frame (described in Section C.3). Here N is the number of points per scan, which is typically 80,000–100,000.
  - *Scan.timestamp_laser*: is the Unix timestamp measured in microseconds since 00:00:00 Jan 1, 1970 UTC for the scan captured by the Velodyne laser scanner.
  - *Scan.timestamp_camera*: is the Unix timestamp measured in microseconds since 00:00:00 Jan 1, 1970 UTC for the closest image (in time) captured by the omnidirectional camera.
  - *Scan.image_index*: is the index of the image that is closest in time to this scan.
  - *Scan.X_wv*: is the 6-DOF pose of the vehicle in the world reference system when the scan was captured.

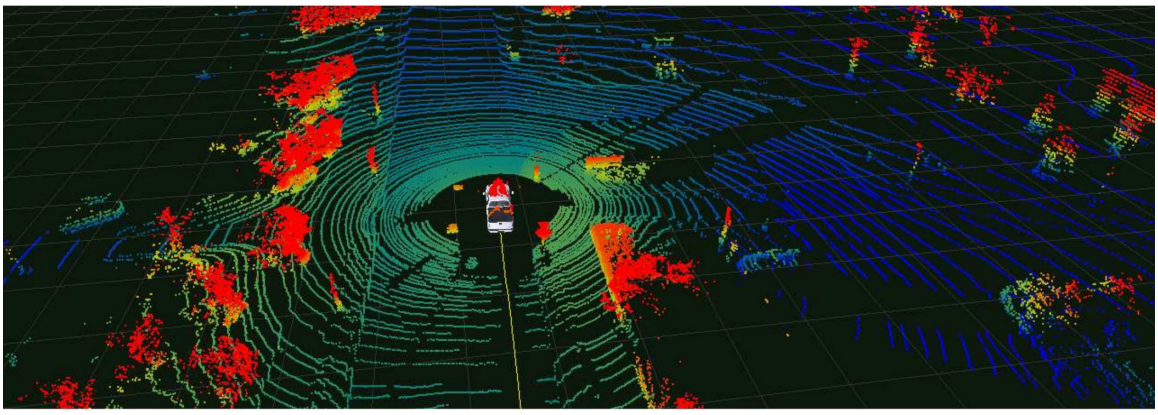
Figure C.6 The left panel shows a spherically distorted image obtained from the Ladybug3 camera. The right panel shows the corresponding undistorted image obtained after applying the transformation provided by the manufacturer.

  - *Scan.Cam*: is a (1×5) array of structures corresponding to each camera of the omni-directional camera system. The format of this structure is given below:
    - *Scan.Cam.points_index*: is a $(1 \times m)$ array of index of the 3D points in laser reference frame, in the field of view of the camera.
    - *Scan.Cam.xyz*: is a $(3 \times m)$ array of 3D laser points $(m < N)$ as represented in the camera reference system and within the field of view of the camera.
    - *Scan.Cam.pixels*: This is a $(2 \times m)$ array of pixel coordinates corresponding to the 3D points projected onto the camera.

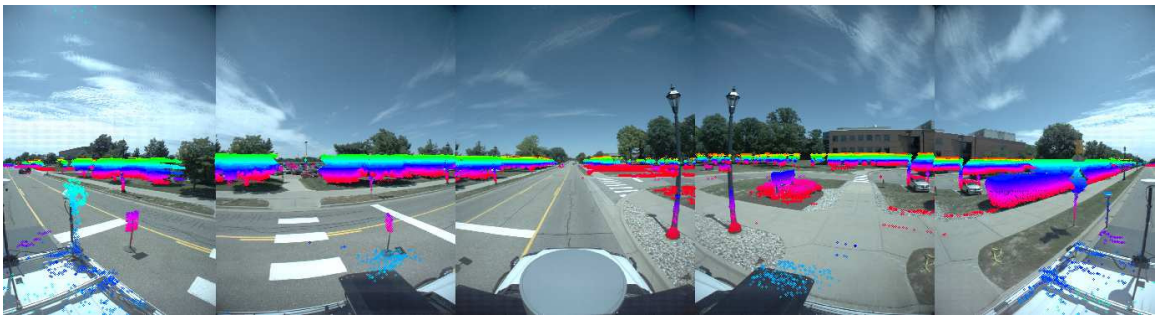
- **VELODYNE**: This folder has several ".pcap" files containing the raw 3D point cloud from the Velodyne laser scanner in a format that can be played back at a desired frame rate using our MATLAB playback tool. This allows the user to quickly browse through the data corresponding to a single trial.

We have provided MATLAB scripts to load and visualize the dataset into the MATLAB workspace. We have tried to keep the data format simple so that it can be easily used by

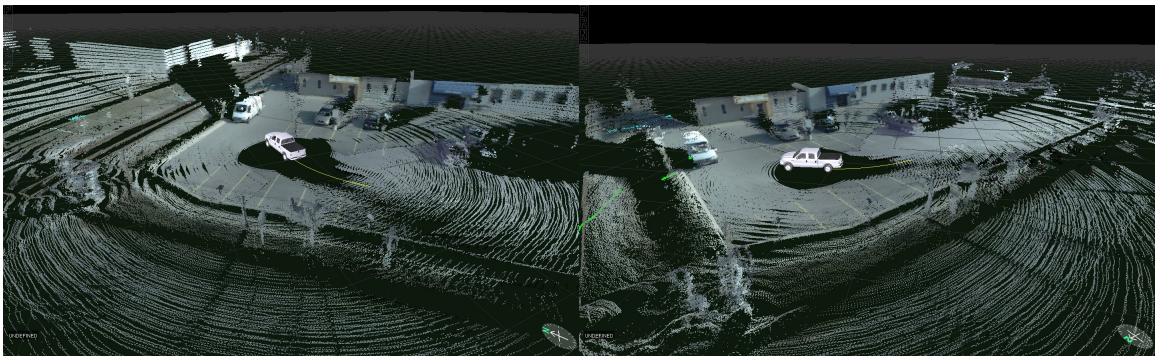
Figure C.7 Reprojection of lidar and camera via extrinsic rigid-body calibration. (a) Perspective view of the 3D lidar range data, color-coded by height above the ground plane. (b) Depiction of the 3D lidar points projected onto the time-corresponding omnidirectional camera image. Several recognizable objects are present in the scene (e.g., people, stop signs, lamp posts, trees). Only nearby objects are projected for visual clarity. (c) Depiction of two different views of a fused lidar/camera textured point cloud. Each 3D point is colored by the RGB value of the pixel corresponding to the projection of the point onto the image.



(a) 3D lidar point cloud



(b) Omnidirectional image with a subset of lidar points projected



(c) Fused RGB textured point cloud

other researchers in their work. We have also provided some visualization scripts (written in MATLAB) with this dataset that allow the re-projection of any scan onto the corresponding omnidirectional imagery as depicted in Fig. C.7. We have also provided a C visualization tool that uses OpenGL to render the textured point cloud as shown in Fig. C.7.

C.5 Notes on data

The time registration between the laser and camera data is not exact due to a transmission offset caused by the $800\ \mathrm{Mb/s}$ Firewire bus over which the camera data is transferred to the computer. The camera data is timestamped as soon as it reaches the computer, so there is a time lag between when the data was actually captured at the camera head and the computer timestamp associated with it. We calculated this approximate time lag (= size of image transferred / transfer rate) and subtracted it from the timestamp of the image to reduce the timing latency.

The *Ford Campus Vision and Lidar dataset* is available for download from our server at `http://robots.engin.umich.edu/SoftwareData/Ford`. Current datasets were collected during November–December 2009, in the future we plan to host more datasets corresponding to some other times of the year.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Agresti, A., and D. Hitchcock (2005), Bayesian inference for categorical data analysis, *Statistical Methods and Applications*, *14*(5), 297–330.

[2] Akca, D. (2007), Matching of 3D surfaces and their intensities, *Journal of Photogrammetry and Remote Sensing*, *62*(2), 112–121.

[3] Alempijevic, A., S. Kodagoda, J. P. Underwood, S. Kumar, and G. Dissanayake (2006), Mutual information based sensor registration and calibration, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 25–30, Orlando, FL, USA.

[4] Alshawa, M. (2007), ICL: Iterative closest line a novel point cloud registration algorithm based on linear features, *Theory and Application of Laser Scanning*, pp. 1–6.

[5] Angeli, A., S. Doncieux, J. Meyer, and D. Filliat (2009), Visual topological SLAM and global localization, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4300–4305, Kobe, Japan.

[6] Ankerst, M., G. Kastenmüller, H. P. Kriegel, and T. Seidl (1999), 3D shape histograms for similarity search and classification in spatial databases, in *Proceedings of the 6th International Symposium on Advances in Spatial Databases*, pp. 207–226, London, UK.

[7] Applanix (2010), POS-LV: Position and orientation system for land vehicles, *Tech. rep.*, Applanix, specification sheets and related articles available at `http://www.applanix.com/products/land/pos-lv.html`.

[8] Artyushkova, K., J. Fenton, J. Farrar, and J. Fulghum (2011), Multitechnique fusion of imaging data for heterogeneous materials, *Image Fusion and Its Applications*, pp. 179–202.

[9] Arun, K. S., T. S. Huang, and S. D. Blostein (1987), Least-squares fitting of two 3D point sets, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *9*(5), 698–700.

[10] Bailey, D. L., D. W. Townsend, P. E. Valk, and M. N. Maisey (2005), *Positron Emission Tomography: Basic Sciences*, Springer-Verlag, Secaucus, NJ.

[11] Bao, S. Y., and S. Savarese (2011), Semantic structure from motion, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2025–2032, Providence, RI, USA.

[12] Barzilai, J., and J. M. Borwein (1988), Two-point step size gradient methods, *IMA Journal of Numerical Analysis*, *8*, 141–148.

[13] Bay, H., T. Tuytelaars, and L. V. Gool (2006), SURF: Speeded up robust features, in *Proceedings of the European Conference on Computer Vision*, pp. 404–417, Graz, Austria.

[14] Beirlant, J., and M. C. A. V. Zuijlen (1985), The empirical distribution function and strong laws for functions of order statistics of uniform spacings, *Journal of Multivariate Analysis*, *16*(3), 300–317.

[15] Belongie, S., and J. Malik (2000), Matching with shape contexts, in *Proceedings of IEEE workshop on Content-based Access of Image and Video Libraries*, pp. 20–26, Santa Barbara, CA, USA.

[16] Berrabah, S. A., H. Sahli, and Y. Baudoin (2011), Visual-based simultaneous localizatiion and mapping and global positioning system correction for geo-localization of a mobile robot, *Measurement Science and Technology*, *22*(12), 233–240.

[17] Besl, P. J., and N. D. McKay (1992), A method for registration of 3D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256.

[18] Biber, P., S. Fleck, and W. Straßer (2004), A probabilistic framework for robust and accurate matching of point clouds, in *Proceedings of 26th Pattern Recognition Symposium (DAGM-04)*, pp. 480–487, Tbingen, Germany.

[19] Boughorbal, F., D. L. Page, C. Dumont, and M. A. Abidi (2000), Registration and integration of multisensor data for photorealistic scene reconstruction, in *Proceedings of 28th AIPR Workshop on 3D Visualization for Data Exploration and Decision Making*, vol. 3905, pp. 74–84, Washington, DC.

[20] Bradski, G. (2000), The OpenCV Library, *Dr. Dobb's Journal of Software Tools*.

[21] Callmer, J., K. Granstrom, J. Nieto, and F. Ramos (2008), Tree of words for visual loop closure detection in urban SLAM, in *Proceedings of the Australasian Conference on Robotics and Automation*, pp. 102–110, Canberra, Australia.

[22] Carlevaris-Bianco, N., A. Mohan, J. R. McBride, and R. M. Eustice (2011), Visual localization in fused image and laser range data, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4378–4385, San Francisco, CA, USA.

[23] Castellanos, J. A., J. Neira, and J. D. Tardos (2002), Multisensor fusion for simultaneous localization and map buildiing., *IEEE Transactions on Robotics and Automation*, *17*, 908–914.

[24] Chao, A., and T. J. Shen (2003), Nonparametric estimation of Shannons index of diversity when there are unseen species in sample, *Environmental and Ecological Statistics*, *10*(4), 429–443.

[25] Chen, Y., and G. Medioni (1992), Object modelling by registration of multiple range images, *Image and Vision Computing*, *10*(3), 145–155.

[26] Chetverikov, D., D. Svirko, D. Stepanov, and P. Krsek (2002), The trimmed iterative closest point algorithm, in *Proceedings of the International Conference on Pattern Recognition*, vol. 3, pp. 545–548, Quebec City, Canada.

[27] Churchill, W., and P. Newman (2012), Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4525–4532, St Paul, MN, USA.

[28] Collignon, A., D. Vandermeulen, P. Suetens, and G. Marchal (1995), 3D multi-modality medical image registration using feature space clustering, *Computer Vision, Virtual Reality and Robotics in Medicine*, *905*, 193–204.

[29] Cover, T. M., and J. A. Thomas (1991), *Elements of Information Theory*, 2^{nd} ed., Wiley Series in Telecommunications and Signal Processing, Hoboken, NJ, USA.

[30] Cramér, H. (1999), *Mathematical methods of statistics*, 9^{th} ed., Princeton University Press, Princeton, NY, USA.

[31] Cummins, M., and P. Newman (2011), Appearance-only SLAM at large scale with FAB-MAP 2.0, *International Journal of Robotics Research*, *30*(9), 1100–1123.

[32] Dudewicz, E. J., and E. C. V. D. Meulen (1981), Entropy based test of uniformity, *Journal of the American Statistical Association*, *76*(376), 967–974.

[33] Ertin, E., J. W. Fisher, and L. C. Potter (2003), Maximum mutual information principle for dynamic sensor query problems, in *2nd International Workshop on Information Processing in Sensor Networks*, pp. 405–416, Springer-Verlag.

[34] Firefly (2010), Imaging products: Firefly IEEE 1394a, *Tech. rep.*, Pointgrey, specification sheet and documentations available at `www.ptgrey.com/products/fireflymv`.

[35] Fischler, M. A., and R. C. Bolles (1981), Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, *24*(6), 381–395.

[36] Fitzgibbon, A. W. (2003), Robust registration of 2D and 3D point sets, *Image and Vision Computing*, *21*(14), 1145–1153.

[37] Forrest, S. (1993), Genetic algorithms: Principles of natural selection applied to computation, *Science*, *261*(5123), 872–878.

[38] Frome, A., D. Huber, R. Kolluri, T. Blow, and J. Malik (2004), Recognizing objects in range data using regional point descriptors, in *Proceedings of the European Conference on Computer Vision*, vol. 3023, pp. 224–237, Prague, Czech Republic.

[39] Gelfand, N., L. Ikemoto, S. Rusinkiewicz, and M. Levoy (2003), Geometrically stable sampling for the ICP algorithm, in *Proceedings of Fourth International Conference on 3D Digital Imaging and Modelling*, pp. 260–267, Banff, Alberta, Canada.

[40] Glover, A. J., W. P. Maddern, M. J. Milford, and G. F. Wyeth (2010), FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3507–3512, Anchorage, Alaska, USA.

[41] Godin, G., D. Laurendeau, and R. Bergevin (2001), A method for the registration of attributed range images, in *Proceeding of International Conference on 3D Digital Imaging and Modeling*, pp. 179–186, Qubec City, Canada.

[42] Gong, X., Y. Lin, and J. Liu (2013), 3D lidar-camera extrinsic calibration using an arbitrary trihedron, *Sensors*, *13*(2), 1902–1918.

[43] Good, I. J. (1965), *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, MIT Press, Cambridge, MA.

[44] Goria, M. N., N. N. Leonenko, V. V. Mergel, and P. L. Noviinverardi (2005), A new class of random vector entropy estimators and its applications in testing statistical hypothesis, *Nonparametric Statistics*, *17*(3), 277–297.

[45] Granger, S., and X. Pennec (2002), Multi-scale EM-ICP: A fast and robust approach for surface registration, in *Proceedings of the European Conference on Computer Vision*, pp. 418–432, London, UK.

[46] Greenspan, M., and M. Yurick (2003), Approximate k-d tree search for efficient ICP, in *Proceedings of Fourth International Conference on 3D Digital Imaging and Modeling*, pp. 442–448, Banff, Alberta, Canada.

[47] Gruen, A., and D. Akca (2004), Least squares 3D surface and curve matching, *Journal of Photogrametry and Remote Sensing*, *59*(3), 151–174.

[48] Gubner, J. A. (2006), *Probability and Random Processes for Electrical and Computer Engineers*, Cambridge University Press, Cambridge, MA, USA.

[49] Hahnel, D., and W. Burgard (2002), Probabilistic matching for 3D scan registration, in *Proceedings of the VDI-Conference Robotik*, Ludwigsburg, Germany.

[50] Hall, P. (1984), Limit theorems for sums of general functions of m-spacings, *Mathematical Proceedings of the Cambridge Philosophical Society*, *96*(3), 517–532.

[51] Hall, P., and S. Morton (1993), On the estimation of entropy, *Annals of the Institute of Statistical Mathematics*, *45*, 1491–1519.

[52] Haralick, R. M. (1998), Propagating covariance in computer vision, in *Proceedings of the Theoretical Foundations of Computer Vision*, pp. 95–114, Dagstuhl, Germany.

[53] Hartley, R., and A. Zisserman (2000), *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, MA, USA.

[54] Hausser, J., and K. Strimmer (2009), Entropy inference and the James-Stein estimator, with application to nonlinear gene association networks, *Journal of Machine Learning Research*, *10*, 1469–1484.

[55] Hendee, W. R., and C. J. Morgan (1984), Magnetic Resonance Imaging Part I – Physical Principles, *Western Journal of Medicine*, *141*(4), 491–500.

[56] Henk, T. (2007), *Understanding Probability: Chance Rules in Everyday Life*, 2^{nd} ed., Cambridge University Press, Cambridge, MA, USA.

[57] Hero, A., and O. Michel (1999), Asymptotic theory of greedy approximations to minimal k-point random graphs, *IEEE Transactions on Information Theory*, *45*(6), 1921–1938.

[58] Hero, A., B. Ma, O. Michel, and J. Gorman (2002), Applications of entropic spanning graphs, *IEEE Signal Processing Magazine*, *19*(5), 85–95.

[59] Hero, A. O., C. M. Kreucher, and D. Blatt (2008), Information theoretic approaches to sensor management, *Foundations and Applications of Sensor Management*, pp. 33–57.

[60] Hill, D., D. Hawkes, N. Harrison, and C. Ruff (1993), A strategy for automated multimodality image registration incorporating anatomical knowledge and imager characteristics, *Information Processing in Medical Imaging*, *687*, 182–196.

[61] Ho, K. L., and P. Newman (2006), Loop closure detection in SLAM by combining visual and spatial appearance, *Robotics and Autonomous Systems*, *54*, 740–749.

[62] Hokuyo (2009), Scanning range finder: UTM-30LX, *Tech. rep.*, Hokuyo, specification sheet and documentations available at `www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html`.

[63] Holste, D., I. Grosse, and H. Herzel (1998), Bayes estimators of generalized entropies, *Journal of Physics*, *31*, 2551–2566.

[64] Horn, B. K. P. (1990), Relative orientation, *International Journal of Computer Vision*, *4*(1), 59–78.

[65] Horvitz, D. G., and D. J. Thompson (1952), A generalization of sampling without replacement from a finite universe., *J. American Statistical Assoc.*, *47*, 663–685.

[66] Huang, A., E. Olson, and D. Moore (2010), LCM: Lightweight communications and marshalling, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4057–4062, Taipei, Taiwan.

[67] Jiang, H. (2002), *Computed Tomography: Principles, Design, Artifacts, and Recent Advances*, SPIE – The International Society for Optical Engineering, Bellingham, Washington.

[68] Joe, H. (1989), On the estimation of entropy and other functionals of a multivariate density, *Annals of the Institute of Statistical Mathematics*, *41*, 683–697.

[69] Johnson, A., and M. Hebert (1999), Using spin images for efficient object recognition in cluttered 3D scenes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *21*(5), 433–449.

[70] Johnson, A., and S. B. Kang (1999), Registration and integration of textured 3D data, *Image and Vision Computing*, *17*, 135–147.

[71] Kadir, T., and M. Brady (2001), Scale, Saliency and Image Description, *International Journal of Computer Vision*, *45*(2), 83–105.

[72] Kaess, M., A. Ranganathan, and F. Dellaert (2008), iSAM: Incremental smoothing and mapping, *IEEE Transactions on Robotics*, *24*(6), 1365–1378.

[73] Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983), Optimization by simulated annealing, *Science*, *220*(4598), 671–680.

[74] Klein, S., M. Staring, and J. Pluim (2007), Evaluation of optimization methods for nonrigid medical image registration using mutual information and B-Splines, *IEEE Transactions on Image Processing*, *16*(12), 2879–2890.

[75] Knops, Z., J. Maintz, M. Viergever, and J. Pluim (2006), Normalized mutual information based registration using k-means clustering and shading correction, *Medical Image Analysis*, *10*(3), 432 – 439, special Issue on The Second International Workshop on Biomedical Image Registration (WBIR).

[76] Konishi, S., A. Yuille, and J. Coughlan (2003), A statistical approach to muli-scale edge detection, *Image and Vision Computing*, *21*(1), 37–48.

[77] Kozachenko, L. F., and N. N. Leonenko (1987), Sample estimate of entropy of a random vector, *Problems of Information Transmission*, *23*, 95–101.

[78] Kramer, G. (1998), Directed information for channels with feedback, in *Dessertation ETH Zrich, Nr. 12656*, Zurich.

[79] Krichevsky, R. E., and V. K. Trofimov (1981), The performance of universal encoding, *IEEE Transactions on Information Theory*, *27*, 199–207.

[80] Krotosky, S. J., and M. M. Trivedi (2007), Mutual information based registration of multimodal stereo videos for person tracking, *Computer Vision and Image Understanding*, *106*, 270–287.

[81] Kullback, S., and R. A. Leibler (1951), On information and sufficiency, *Anals of Mathematical Statistics*, *22*(1), 79–86.

[82] Ladybug3 (2009), Spherical vision products: Ladybug3, *Tech. rep.*, Pointgrey, specification sheet and documentations available at `www.ptgrey.com/products/ladybug3/index.asp`.

[83] Lazebnik, S., C. Schmid, and J. Ponce (2005), A sparse texture representation using local affine regions, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *27*, 1265–1278.

[84] Learned-Miller, E. G. (2003), A new class of entropy estimators for multi-dimensional densities, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP03)*, vol. 3, pp. 297–300, Hong Kong, China.

[85] Lee, I. (2010), Sample-spacings based density and entropy estimators for spherically invariant multidimensional data, *Neural Computation*, *22*(8), 2208–2227.

[86] Lehmann, E. L., and G. Casella (2011), *Theory of Point Estimation*, Springer Texts in Statistics Series, Springer, New York, NY, USA.

[87] Leonard, J., et al. (2007), Team MIT Urban Challenge technical report, *Tech. rep.*, Massachusetts Institute of Technology.

[88] Leonenko, N. N., L. Prozanto, and V. Savani (2008), A class of Renyi information estimators for multidimensional densities, *Annals of Statistics*, *36*, 2153–2182.

[89] Levenberg, K. (1944), A method for the solution of certain problems in least squares, *The Quarterly of Applied Mathematics*, *2*, 164–168.

[90] Levinson, J., and S. Thrun (2010), Unsupervised calibration for multi-beam lasers, in *Proceedings of the International Conference on Experimental Robotics*, Delhi, India.

[91] Levinson, J., and S. Thrun (2012), Automatic calibration of cameras and lasers in arbitrary scenes, in *Proceedings of the International Conference on Experimental Robotics*, Quebec City, Canada.

[92] Li, G., Y. Liu, L. Dong, X. Cai, and D. Zhou (2007), An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3854–3859, San Diego, CA, USA.

[93] Li, Y., Y. Ruichek, and C. Cappelle (2013), Optimal extrinsic calibration between a stereoscopic system and a lidar, *IEEE Transactions on Instrumentation and Measurement*, *62*(8), 2258–2269.

[94] Lin, Y., and G. Medioni (2008), Mutual information computation and maximization using GPU, in *Computer Vision and Pattern Recognition Workshop*, pp. 1–6, Anchorage, Alaska, USA.

[95] Lindley, D. V. (1964), The Bayesian analysis of contingency tables, *The Annals of Mathematical Statistics*, *35*, 1622–1643.

[96] Lowe, D. G. (2004), Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, *60*(2), 91–110.

[97] Luan, H., F. Qi, Z. Xue, L. Chen, and D. Shen (2008), Multimodality image registration by maximization of quantitative-qualitative measure of mutual information, *Pattern Recognition*, *41*, 285–298.

[98] Maes, F., A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens (1997), Multimodality image registration by maximization of mutual information, *IEEE Transactions on Medical Imaging*, *16*, 187–198.

[99] Marquadrt, D. (1963), An algorithm for least-squares estimation of nonlinear parameters, *SIAM Journal on Applied Mathematics*, *11*, 431–441.

[100] Massey, J. L. (1990), Causality, feedback and directed information, in *International Symposium on Information Theory and its Applications*, pp. 303–305, Hawaii, USA.

[101] McBride, J. R., J. C. Ivan, D. S. Rhode, J. D. Rupp, M. Y. Rupp, J. D. Higgins, D. D. Turner, and R. M. Eustice (2008), A perspective on emerging automotive safety applications, derived from lessons learned through participation in the DARPA grand challenges, *Journal of Field Robotics*, *25*(10), 808–840.

[102] McDonald, J. H. (2009), *Handbook of Biological Statistics*, 2^{nd} ed., Sparky House Publishing, Baltimore, MD USA.

[103] Mei, C., and P. Rives (2006), Calibration between a central catadioptric camera and a laser range finder for robotic applications, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 532–537, Orlando, FL, USA.

[104] Milford, M., G. Wyeth, and D. Prasser (2004), RatSLAM: A hippocampal model for simultaneous localization and mapping., in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 403–408, Barcelona, Spain.

[105] Milford, M. J., and G. F. Wyeth (2012), SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1643–1649, St. Paul, MN, USA.

[106] Miller, G. (1955), Note on the bias of information estimates, *Information Theory in Psychology: Problems and Methods*, *2*, 95–100.

[107] Mills, D. (2006), Network time protocol version 4 reference and implementation guide, *Tech. Rep. 06-06-1*, University of Delaware.

[108] Mirzaei, F. M., D. G. Kottas, and S. I. Roumeliotis (2012), 3D lidar-camera intrinsic and extrinsic calibration: Observability analysis and analytical least squares-based initialization, *International Journal of Robotics Research*, *31*(4), 452–467.

[109] Mitra, N. J., N. Gelfand, H. Pottmann, and L. Guibas (2004), Registration of point cloud data from a geometric optimization perspective, in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 22–31.

[110] Moghadam, P., M. Bosse, and R. Zlot (2013), Line-based extrinsic calibration of range and image sensors, in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 4–11, Karlsruhe, Germany.

[111] Montemerlo, M., et al. (2008), Junior: The Stanford Entry in the Urban Challenge, *Journal of Field Robotics*, *25*(9), 569–597.

[112] Montesano, L., J. Minguez, and L. Montano (2005), Probabilistic scan matching for motion estimation in unstructured environments, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3499–3504, Edmonton, Alberta, Canada.

[113] Moore, D., A. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller (2009), Simultaneous local and global state estimation for robotic navigation, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3794–3799, Kobe, Japan.

[114] Napier, A., P. Corke, and P. Newman (2013), Cross-calbration of push-broom 2D lidars and cameras in natural scenes, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3664–3669, Karlsruhe, Germany.

[115] Nelder, J. A., and R. Mead (1965), A simplex method for function minimization, *The Computer Journal*, *7*(4), 308–313.

[116] Nemenman, I., F. Shafee, and W. Bialek (2002), Entropy and inference, revisited, *Advances in Neural Information Processing Systems*, *14*, 471–478.

[117] Neubert, P., N. Sunderhauf, and P. Protzel (2013), Appearance change prediction for long-term navigation across seasons, in *Proceedings of the 4th European Conference on Mobile Robots*, Barcelona, Spain, Accepted, To Appear.

[118] Newman, P., D. Cole, and K. Ho (2006), Outdoor SLAM using visual appearance and laser ranging, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1180–1187, Orlando, FL, USA.

[119] Nie, Y., Q. Chen, T. Chen, Z. Sun, and B. Dai (2012), Camera and lidar fusion for road intersection detection, in *IEEE Symposium on Electrical and Electronics Engineering (EEESYM)*, pp. 273–276, Kuala Lumpur, Malaysia.

[120] Nister, D., and H. Stewenius (2006), Scalable recognition with a vocabulary tree, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2161–2168, New York, NY, USA.

[121] Nunnez, P., P. D. Jr., R. Rocha, and J. Dias (2009), Data fusion calibration for a 3D laser range finder and a camera using inertial data, in *Proceedings of the 4th European Conference on Mobile Robots*, pp. 31–36, Mlini/Dubrovnik, Croatia.

[122] Olson, E. (2010), A passive solution to the sensor synchronization problem, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1059–1064, Taipei, Taiwan.

[123] Olson, E. B. (2009), Real-time correlative scan matching, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1233–1239.

[124] Opegen-Rhein, R., and K. Strimmer (2007), Accurate ranking of differentially expressed genes by a distribution-free shrinkage approach, *Statistical Applications in Genetics and Molecular Biology*, *6*(9).

[125] Orlitsky, A., N. P. Santhanam, and J. Zhang (2003), Always good turing: Asymptotically optimal probability estimation, *Science*, *302*, 427–431.

[126] Pandey, G., J. R. McBride, S. Savarese, and R. M. Eustice (2010), Extrinsic calibration of a 3d laser scanner and an omnidirectional camera, in *IFAC Symposium on Intelligent Autonomous Vehicles*, vol. 7, pp. 336–341, Leece, Italy.

[127] Panzeri, S., and A. Treves (1996), Analytical estimates of limited sampling biases in different information measures, *Computation in Neural Systems*, *7*, 87–107.

[128] Park, S.-Y., and M. Subbarao (2003), An accurate and fast point-to-plane registration technique, *Pattern Recognition Letters*, *24*(16), 2967–2976.

[129] Paul, R., and P. Newman (2010), FAB-MAP 3D: Topological mapping with spatial and visual appearance, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2649–2656, Anchorage, Alaska, USA.

[130] Pierce, J. R. (1980), *An Introduction to Information Theory: Symbols, Signals and Noise*, Dover Publications.

[131] Pluim, J. P. W., J. B. A. Maintz, and M. A. Viergever (), Mutual information matching in multi-resolution contexts, *Image and Vision Computing*, *19*(1), 45–52.

[132] Pluim, J. P. W., J. B. A. Maintz, and M. A. Viergever (2003), Mutual information based registration of medical images: A survey, *IEEE Transactions on Medical Imaging*, *22*(8), 986–1004.

[133] Premebida, C., O. Ludwig, and U. Nunes (2009), Lidar and vision-based pedestrian detection system, *Journal of Field Robotics*, *26*(9), 696–711.

[134] Pronobis, A., B. Caputo, P. Jensfelt, and H. I. Christensen (2006), A discriminative approach to robust visual place recognition, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3829–3836, Beijing, China.

[135] Renyi, A. (1960), On measures of information and entropy, in *Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability*, pp. 547–561, Berkeley, CA, USA.

[136] Riegl (2010), LMS-Q120: 2D laser scanner, *Tech. rep.*, Riegl, specification sheet and documentations available at `http://www.riegl.com/nc/products/mobile-scanning/produktdetail/product/scanner/14`.

[137] Rodriguez, F., V. Fremont, P. Bonnifait, et al. (2008), Extrinsic calibration between a multi-layer lidar and a camera, in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 214–219, Seoul, South Korea.

[138] Ross, S. (2009), *First Course in Probability*, 8^{th} ed., Prentice Hall, Upper Saddle River, NJ, USA.

[139] Rusinkiewicz, S., and M. Levoy (2001), Efficient variants of the ICP algorithm, in *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, Quebec City, Canada.

[140] Rusu, R. B., and S. Cousins (2011), 3D is here: Point cloud library (PCL), in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1–4, Shanghai, China.

[141] Rusu, R. B., N. Blodow, and M. Beetz (2009), Fast point feature histograms (FPFH) for 3D registration, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3212–3217, Kobe, Japan.

[142] Scaramuzza, D., A. Harati, and R. Siegwart (2007), Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4164–4169, San Diego, CA, USA.

[143] Schafer, J., and K. Strimmer (2005), A shrinkage approach to large scale covariance matrix estimation implications for functional genomics, *Statistical Applications in Genetics and Molecular Biology*, *4*(32).

[144] Schurmann, T., and P. Grassberger (1996), Entropy estimation of symbol sequences, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *6*(3), 414–427.

[145] Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, John Wiley, New York.

[146] Segal, A. V., D. Haehnel, and S. Thrun (2009), Generalized-ICP, in *Proceedings of the Robotics: Science & Systems Conference*, Seattle, WA, USA.

[147] Shams, R., P. Sadeghi, and R. Kennedy (2007), Gradient intensity: A new mutual information-based registration method, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA.

[148] Shams, R., P. Sadeghi, R. Kennedy, and R. Hartley (2010), Parallel computation of mutual information on the GPU with application to real-time registration of 3D medical images, *Computer Methods and Programs in Biomedicine*, *99*(2), 133–146.

[149] Shannon, C. E. (1948), A mathematical theory of communication, *Bell Systemm Technical Journal*, *27*(3), 50–64.

[150] Silverman, B. W. (1986), Density estimation for statistics and data analysis, *Monographs on Statistics and Applied Probability*.

[151] Singh, H., N. Misra, V. Hnizdo, A. Fedorowicz, and E. Demchuk (2003), Nearest neighbour estimators of entropy, *American Journal of Mathematical and Management Sciences*, *23*(3), 301–321.

[152] Sivic, J., and A. Zisserman (2003), Video google: A text retrieval approach to object matching in videos, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1470–1477, Nice, France.

[153] Smith, R., M. Self, and P. Cheeseman (1988), A stochastic map for uncertain spatial relationships, in *Proceedings of the International Symposium on Robotics Research*, pp. 467–474, Santa Clara, CA, USA.

[154] Sricharan, K., R. Raich, and A. O. Hero (2011), K-nearest neighbor estimation of entropies with confidence, in *IEEE International Symposium on Information Theory*, pp. 1205–1209, Saint Peterusberg, Russia.

[155] Staring, M., U. Van der Heide, S. Klein, M. Viergever, and J. P. W. Pluim (2009), Registration of cervical mri using multifeature mutual information, *IEEE Transactions on Medical Imaging*, *28*(9), 1412–1421.

[156] Steder, B., R. B. Rusu, K. Konolige, and W. Burgard (2010), NARF: 3D range image features for object recognition, in *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan.

[157] Studholme, C., D. Hill, and D. Hawkes (1996), Automated 3-D registration of MR and CT images of the head, *Medical Image Analysis*, *1*(2), 163–175.

[158] Studholme, C., D. L. G. Hill, and D. J. Hawkes (1999), An overlap invariant entropy measure of 3d medical image alignment, *Pattern Recognition*, *32*(1), 71–86.

[159] Sunderhauf, N., and P. Protzel (2011), BRIEF-Gist—Closing the loop by simple means, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1234–1241, San Fransisco, CA, USA.

[160] Tamjidi, A., and C. Ye (2012), 6-DOF pose estimation of an autonomous car by visual feature correspondence and tracking, *International Journal of Intelligent Control and Systems*, *17*(3), 94–101.

[161] Tsybakov, A. B., and E. C. V. D. Meulen (1996), Root-n consistent estimators of entropy for densities with unbounded support, *Scandinavian Journal of Statistics*, *23*(1), 75–83.

[162] Unal, G., A. Yezzi, and H. Krim (2005), Information-theoretic active polygons for unsupervised texture segmentation, *International Journal of Computer Vision*, *62*(3), 199–220.

[163] Unnikrishnan, R., and M. Hebert (2005), Fast extrinsic calibration of a laser rangefinder to a camera, *Tech. Rep. CMU-RI-TR-05-09*, Robotics Institute Carnegie Mellon University.

[164] Vasicek, O. (1976), A test for normality based on sample entropy, *Journal of the Royal Statistical Society*, *38*(1), 54–59.

[165] Velodyne (2007), Velodyne HDL-64E: A high definition LIDAR sensor for 3D applications, *Tech. rep.*, Velodyne, available at `www.velodyne.com/lidar/products/white_paper`.

[166] Viola, P. A., and W. M. Wells (1997), Alignment by maximization of mutual information, *International Journal of Computer Vision*, *24*, 137–154.

[167] Whittaker, E. T., and G. Robinson (1967), The Newton-Raphson method, *The Calculus of Observations: A Treatise on Numerical Mathematics*, *44*(4), 84–87.

[168] Williams, N., K. L. Low, C. Hantak, M. Pollefeys, and A. Lastra (2004), Automatic image alignment for 3D environment modeling, in *Proceedings of IEEE Brazilian Symposium on Computer Graphics and Image Processing*, pp. 388–395, Curitiba, PR, Brazil.

[169] Woods, R. P., S. R. Cherry, and J. C. Mazziotta (1992), Rapid automated algorithm for aligning and reslicing PET images, *Journal Of Computer Assisted Tomography*, *16*(4), 620–633.

[170] Woods, R. P., S. R. Cherry, and J. C. Mazziotta (1993), MRI-PET registration with automated algorithm, *Journal Of Computer Assisted Tomography*, *17*(4), 536–546.

[171] Xsens (2010), MTi-G a GPS aided MEMS based Inertial Measurement Unit (IMU) and static pressure sensor, *Tech. rep.*, Xsens, specification sheet and documentations available at `http://www.xsens.com/en/general/mti-g`.

[172] Xu, Z., R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck (2005), Smart pixel — photonic mixer device (PMD) new system concept of a 3D-imaging camera-on-a-chip, *Tech. rep.*, PMD Technologies.

[173] Zhang, Q. (2004), Extrinsic calibration of a camera and laser range finder, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2301–2306, Sendai, Japan.

[174] Zhang, Z. (2000), A flexible new technique for camera calibration, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(11), 1330–1334.

[175] Zhou, L., and Z. Deng (2012), Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation, in *Proceedings of IEEE Intelligent Vehicles Symposium (IV)*, pp. 642–648, Madrid, Spain.