論文 / 著書情報
Article / Book Information

題目(和文)	統計的依存性を利用した移動ロボットのナビゲーション
Title(English)	Mobile Robot Navigation Using Statistical Dependence
著者(和文)	入江清
Author(English)	Kiyoshi Irie
出典(和文)	学位:博士（工学）, 学位授与機関:東京工業大学, 報告番号:甲第10351号, 授与年月日:2016年9月20日, 学位の種別:課程博士, 審査員:小池 英樹,篠田 浩一,石田 貴士,下坂 正倫,鈴木 大慈,杉山 将
Citation(English)	Degree:, Conferring organization: Tokyo Institute of Technology, Report number:甲第10351号, Conferred date:2016/9/20, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Thesis

Mobile Robot Navigation Using Statistical Dependence

Author:
Kiyoshi Irie

Supervisors:
Hideki Koike
Masashi Sugiyama

Department of Computer Science
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Engineering

August 2016

Contents

List of Figures

List of Tables

Abstract

Navigation is an important capability for mobile robots and has a variety of application such as autonomous driving vehicles and service tasks. Although we humans perform navigation daily and easily, no robot has been developed yet that can arbitrarily navigate in unrehearsed, complex real environments. For example, to operate current navigation systems, we firstly need to construct a detailed map of the target environment by manually navigating the robot and collecting data.

This thesis is devoted to extending robots' navigation ability aiming to realize navigation in unrehearsed environments. We have found that statistical dependence is a promising tool for the problem. We explored applications of statistical dependence to unresolved navigation issues and developed two new methods of: a) street-map-based localization and b) target-less multi-modal sensor extrinsic calibration.

First, we consider the problem of localization based on a 2D street map. Localization, a process of estimating the position and orientation of a mobile robot with respect to a map, is one of the essential aspects of navigation. The current state-of-the-art methods localize a robot on a detailed 3D map, which is typically constructed by a simultaneous localization and mapping (SLAM) technique. However, the construction cost of such a map dedicated to robot increases rapidly as the expansion of the target environment. We aim to overcome the problem by reusing 2D street maps that people use daily. One of the major challenges in using 2D street maps is finding the correct correspondence in completely different information, that is, the map and sensor data collected by the robot.

We propose a novel localization method, dependence maximization localization, which is inspired by multi-modal image registration. The localization problem is formulated as registration between sensor data and the map. Registration is performed through maximization of statistical dependence measured by least-squares mutual information (LSMI).

We also applied our method to a position tracking problem (i.e., time-series localization of a moving robot). A particle filter that fuses the robot's motion estimation and observation is employed. In the update step of the particle filter, we compute LSMI between sensor data and the map for each particle. The particles are then weighted and resampled according to the LSMI values.

Extensive experiments were conducted using data sets collected in real urban environments, and we empirically found out that LSMI exhibits superior performance among several dependence estimators. Successful results obtained for long distance (up to 4.5 km) position tracking experiments support the efficacy of the proposed method.

Second, we consider the problem of extrinsic calibration (identifying the relative pose) between multi-modal sensors. In the localization experiments mentioned above, a combination of a camera and a light detection and ranging (LiDAR) sensor is employed. Such a sensor combination is a popular means of acquiring colored point clouds and widely used for outdoor mobile robots. Automatic recalibration of the sensors is important for maintaining the quality of the colored points because the sensor configurations can change over time. Conventional calibration methods employ special calibration targets such as a board with a checker pattern; however, relying on such a calibration target can hinder the automation of the calibration process. Therefore, calibration in natural scenes (so-called target-less calibration) is receiving increasing interest from the robotics community.

We propose a novel method for target-less extrinsic calibration between a camera and a LiDAR. Although there exist several calibration methods of the same purpose using statistical dependence, they are found to perform poorly under strong sunlight. The cause of the problem was that the employed features and dependence estimators were susceptible to noise; the objective functions employed in the previous methods become non-smooth on noisy data and make gradient-based search ineffective. Using richer features is necessary to overcome the problem; however, the dependence estimators employed in the previous methods are not capable of handling high-dimensional features. Therefore, we develop a novel dependence estimator, bagged least-squares mutual information (BLSMI), which can robustly estimate dependence of high-dimensional data.

BLSMI is a combination of bootstrap aggregating (bagging) and LSMI

which is a kernel-based, robust dependence estimator. It is demonstrated that the combination provides highly smooth objective function even on noisy data. We tested our method using real-world data sets collected in both indoor and outdoor environments. Our proposed method outperformed existing methods in terms of calibration accuracy. The advantage of our method in accuracy was particularly significant on outdoor data sets.

We conclude that these successful applications of statistical dependence form a significant contribution to the study of mobile robot navigation and worth a further study in the future.

Acknowledgements

First of all, I am deeply indebted to my academic advisor, Professor Masashi Sugiyama, for his enthusiastic guidance and an excellent research environment. He always provided me with helpful and prompt suggestions in spite of his extremely busy schedule. Also, I would like to express my sincere gratitude to Dr. Masahiro Tomono, for his support and insightful comments. Without his support, I could not even start my carrier as a researcher.

Many thanks to Tomoya Sakai, Voot Tangkaratt, Ikko Yamane, Hideko Kawakubo, Hiroaki Shiino, and all the members of Sugiyama-Sato laboratory for taking time to discuss with me.

I cannot forget to acknowledge the support of my boss at Chiba Institute of Technology (CIT), Dr. Takyuki Furuta, who allowed me to take my doctoral program while working full-time at CIT.

I would also like to recognize the support and encouragement of my colleagues at CIT, my parents, my brothers and my sisters.

A portion of this research was funded by the New Energy and Industrial Technology Development Organization (NEDO).

Chapter 1

Introduction

Navigation is a fundamental functionality of mobile robots [Oommen *et al.*, 1987] [Kobayashi *et al.*, 2001] [Kümmerle *et al.*, 2013]. Although humans perform navigation daily and easily, we have not developed a robot that can arbitrarily navigate in unrehearsed, complex, and real environments. This thesis is devoted to extending robots' ability of navigation by addressing specific unresolved issues.

1.1 Mobile Robot Navigation

Navigation is an indispensable functionality for various robot applications such as luggage carrying [Misawa *et al.*, 2007], autonomous driving of cars [Jochem *et al.*, 1995] and personal mobility [Hatao *et al.*, 2009], and tour-guiding [Thrun *et al.*, 1999]. In the last several decades, there has been significant progress in navigation research [Thorpe *et al.*, 1988] [Langer and Thorpe, 1992] [Montemerlo *et al.*, 2008] [Kümmerle *et al.*, 2013]. Nowadays, indoor robot navigation has progressed, and we can see cleaning robots with navigation functionality in the market (Figure 1.1 (a)) [iRobot, 2016] [Dyson, 2016]. In addition, several universities and manufacturers are testing their self-driving cars in real environments (Figure 1.1 (b)) [Thrun, 2010] [Guizzo, 2011]. However, certain navigation issues remain unresolved, particularly those regarding *outdoor* environments. Here, we briefly review the advancement of navigation research.

Studies on outdoor navigation began more than 30 years ago. One of the early systems was Navlab [Thorpe *et al.*, 1988], which was equipped with a camera and a laser range finder; Navlab was able to detect and following roads. Since then, intensive research has been conducted on long-distance

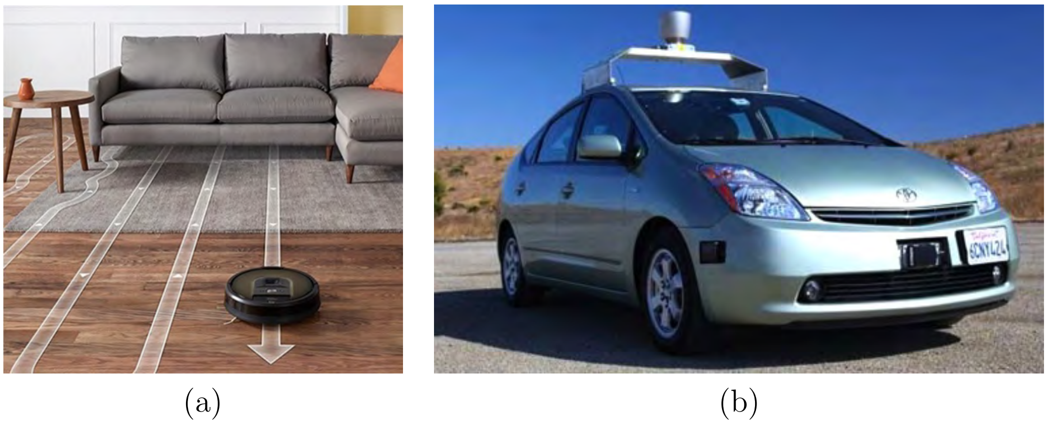

(a) (b)

FIGURE 1.1: Mobile robot systems expected to work closely to our daily life. (a) House cleaning robot equipped with navigation ability [iRobot, 2016]. (b) Google's self-driving car [Guizzo, 2011].

navigation. In the early 1990s, Navlab 5 [Jochem *et al.*, 1995] was reported as the system that had driven 2849 miles. However, its navigation was not fully autonomous; 1.8% manual steering was required [Bertozzi *et al.*, 2000]. In the DARPA Grand Challenge held in 2005, autonomous robots competed in driving on unrehearsed off-road terrain. Five robots successfully navigated the 175 miles pathway [DARPA, 2007]. Advancement of on-road navigation was demonstrated in the DARPA Urban Challenge which was held in 2007, when six robots completed on 60 miles fully autonomous navigation in a simulated urban environment [Buehler *et al.*, 2009]. Despite the successful results, the technology employed in the challenges was not sufficient for real environments. One of the key enabling components for real-world navigation is precise *localization*.

Robot localization, that is estimating the position and orientation of the robot, is one of the essential aspects of navigation. Poor localization accuracy can cause many problems, e.g., driving out of the road and incorrect object recognition. In the DARPA challenges, the navigation route was given by the global positioning systems (GPS) coordinates; therefore, the robots employed GPS combined with an inertial navigation system to estimate their position [Montemerlo *et al.*, 2008]. Although the accuracy of GPS is not very high (typical error is more than 1 m), the roads used for the challenges were wide for this degree of precision [Levinson and Thrun, 2010]. However, navigation in complex real environments requires higher localization accuracy beyond the

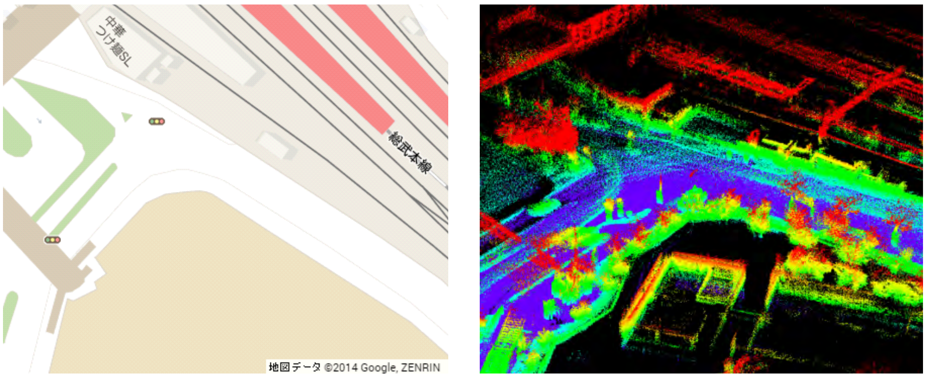


FIGURE 1.2: Comparison of maps for human and robots. Left: sample 2D street map provided by Google Maps. Right: sample 3D map constructed by, and for a robot.

limitation of GPS.

The invention of wide-range 3D light detection and ranging (LiDAR) sensors and simultaneous localization and mapping (SLAM) [Thrun *et al.*, 2005] opened up the route to a different localization approach. The combination of LiDARs and SLAM made it possible to construct large-scale detailed maps. Figure 1.2 provides a comparison of a street map for humans and one for robots. As can be seen from the figure, highly detailed 3D maps can be obtained by SLAM, though we humans can navigate without such a detailed map. Using such a detailed map, robot localization with sub-meter accuracy was achieved, and navigation in real urban environments became possible [Levinson and Thrun, 2010]. This is the current state-of-the-art in localization for navigation and currently, many mobile robots and self-driving cars including Google's one employ the same localization approach [Yoshida *et al.*, 2010] [Kümmerle *et al.*, 2013] [Guizzo, 2011]. However, navigation using this localization method is restricted to environments, whose maps are prebuilt. This means that we need costly preparation before applying the system to new environments: e.g., collecting data with manually navigated robots, constructing the map, and confirming the map's accuracy. In other words, the current navigation systems cannot be used in *unrehearsed* environments.

1.2 Research Objective and Our Approach

The objective of this thesis is to extend robots' ability toward navigation in unrehearsed environments. While navigation has several technical aspects such as localization, path planning, and obstacle recognition, this thesis considers two issues, namely, street-map-based localization and extrinsic calibration of multi-modal sensors. We address both issues using a single framework, by registering sensor data using statistical dependence. Although statistical dependence has been employed for various problems [Pluim *et al.*, 2003] [Peng *et al.*, 2005] [Alempijevic *et al.*, 2009] [Pandey *et al.*, 2014] [Pethel and Hahs, 2014], its usefulness for navigation has not been fully explored yet.

Here, we discuss the background and importance of the two issues. We also show that the two issues share similar structure and how our approach based on statistical dependence is applied to these issues.

1.2.1 Street-map-based Localization

We first consider the localization problem because map construction and maintenance is one of the most fundamental issues to be addressed for realizing navigation in unrehearsed environments. In this dissertation, we propose to reuse existing 2D street maps for localization. Nowadays digital street maps (e.g., those provided by Google Maps [1]) are widely available and used routinely. By reusing such maps, we can eliminate the cost for construction and maintenance of maps dedicated to robots. Although localization can be regarded as a registration problem between the robot's sensor observation and the map, registering sensor data on a street map is not straightforward. This is because street maps contain completely different information than sensor data, and no detailed information is available (Figure 1.2). For example, a street map does not contain information on the colors and shapes of the objects in the environment. That makes it difficult to find a correspondence between sensor data and a map.

We have found out that statistical dependence can be effectively applied

---

[1] `https://maps.google.com/`

to overcome this problem. Statistical dependence has been used for multi-modal image registration [Wells *et al.*, 1996] [Pluim *et al.*, 2003] [Alempijevic *et al.*, 2009] [Mastin *et al.*, 2009]; correspondence between two images of different modalities can be made through maximization of statistical dependence. Dependence-based image registration can be performed, without using detailed knowledge of the target object appearing in the images. We take advantage of the characteristics of the statistical dependence and apply it to the localization problem.

Based on the idea mentioned above, we propose a novel localization method. Our approach is to maximize statistical dependence between a map and features extracted from observed sensor data. By using statistical dependence, we can register sensor data collected from unknown target objects with a street map and estimate the robot's position. Squared-loss mutual information (SMI) [Suzuki *et al.*, 2009a], which is a robust statistical dependence measure, is employed in our method. Robustness is important, because data to be registered are noisy; sensor observations contain noises and street map often include errors.

The advantage of our approach is versatility. Although most existing street-map-based localization methods focus on self-driving cars and assume the existence of lane markings or curb-like obstacles at the road boundaries [Chausse *et al.*, 2005] [Morales *et al.*, 2009], these assumptions are easily violated in environments such as sidewalks. Instead, the statistical dependence approach relaxes this assumption and enables localization in environments that the existing methods cannot handle. Therefore, our approach is applicable to not only self-driving cars but also various types of robots and mobilities.

1.2.2 Multi-modal Sensor Extrinsic Calibration

Registration using statistical dependence can be applied to another navigation-related problem, multi-modal sensor extrinsic calibration. A combination of multi-modal sensors (e.g., a camera and LiDAR) is widely used in mobile robots [Thorpe *et al.*, 1988] [Montemerlo *et al.*, 2008] [Kümmerle *et al.*, 2013]. In order to combine information from different sensors, knowing the geometric relationship between them is useful. Extrinsic calibration of sensors is a problem of estimating the relative position and orientation between sensors.

Automatic calibration and recalibration are important because sensor configuration can change over time.

Sensor extrinsic calibration can be achieved by registering the data obtained from each sensor. However, when the sensors are of different modalities, their registration is challenging because different types of information need to be registered. For example, a stereo camera system, which is a combination of sensors of the same modality, can be calibrated by merely registering images using image local features. However, such a simple technique cannot be directly applied to LiDAR data; therefore, calibration between camera and LiDAR is not straightforward [Mastin *et al.*, 2009] [Li and Olson, 2010].

Various calibration methods have been explored to overcome the difficulty of registration. For example, Scaramuzza *et al.* [Scaramuzza *et al.*, 2007] proposed a method in which the correspondences were manually provided by a human operator. Alternatively, some methods employed calibration targets designed to enable easy finding of correspondences [Zhang and Pless, 2004] [Kwak *et al.*, 2011] [Alismail *et al.*, 2012] [Geiger *et al.*, 2012]. A typical choice for the calibration target is a board with a checker pattern. However, relying on such special targets makes it difficult to achieve fully automated calibration. In addition, because we are aiming to perform calibration in unrehearsed environments, we cannot assume the availability of such calibration targets.

Based again on statistical dependence, we develop a novel method for extrinsic calibration of a camera-LiDAR system. The relative pose between the two sensors is estimated through maximization of squared-loss mutual information. As we have already discussed above, statistical dependence-based registration is applicable to sensor data obtained from unknown targets. Therefore, our calibration approach could be applied in unrehearsed environments.

Although there are several calibration methods using statistical dependence, they are ineffective under strong sunlight. The cause of this problem was in the non-smoothness of the dependence estimator employed in the methods. Because of the non-smoothness, gradient-based search can easily get trapped in local optima. We found that richer features had to be incorporated to address this problem; however, the previously employed dependence estimators [Pandey *et al.*, 2012] could not handle higher dimensional features. To overcome this difficulty, we will develop a novel dependence estimator.

1.3 Contribution

Here, we summarize the main contribution of this thesis.

1.3.1 A Novel Approach to Street-map-based Localization

We found out that localization is possible by maximizing the statistical dependence between the sensor data and the map. To our knowledge, statistical dependence has never been applied to street-map-based localization. Our method enables localization in urban sidewalk environments, for which existing methods are not effective. Furthermore, we demonstrate the importance of the robustness of a dependence estimator. We discovered that, among several different dependence measures, SMI performs best in terms of localization accuracy.

1.3.2 Application to Position Tracking Problem

We extend the abovementioned localization method to time-series position estimation of a moving robot (position tracking). We propose fusing the robot's motion estimation and observation based on SMI estimation, using a particle filter. A computationally efficient implementation is also presented, in which we achieved near real-time position tracking performance on a laptop-class computer. Through extensive experiments using real-world datasets, it is demonstrated that our method can robustly track the robot's position.

1.3.3 A Low-variance and Robust Statistical Dependence Estimator

We develop a novel SMI estimator by extending least-squares mutual information (LSMI) [Sakai and Sugiyama, 2014], which is used for the localization problem mentioned above. Although LSMI has been successfully applied to the localization problem, we observed that maximization of LSMI using its numerical gradient sometimes results in local optima. Here, we solve this problem by reducing the variance of LSMI. We propose combining bagging [Breiman, 1996] with LSMI for low-variance SMI estimation. We refer to our novel estimator as *bagged least-squares mutual information (BLSMI)* and apply it to the calibration problem described below.

1.3.4 A Novel Multi-modal Sensor Extrinsic Calibration Method

We propose a novel target-less extrinsic calibration method robust to illumination conditions. The robustness is achieved by introducing BLSMI and incorporating richer features. The performance of the developed method is tested through experiments using real-world datasets. It is demonstrated that our method exhibits superior calibration accuracy compared with several existing methods.

1.4 Organization

The reminder of this thesis is organized as follows (Figure 1.3).

Chapter 2 presents an overview of statistical dependence and its estimators. Several dependence measures including mutual information (MI) and its variants are discussed. SMI and its estimators, which are used as primary tools in later chapters, are highlighted. We introduce a novel dependence estimator called BLSMI for robust and low-variance estimation of SMI.

Two main the contributions are presented in Chapters 3 and 4. Chapter 3 presents the first successful application of statistical dependence to street-map-based localization. While most conventional methods employ road boundary detection, they are not applicable to complex environments such as sidewalks. We show that localization in real urban environments is possible through maximization of statistical dependence.

Chapter 4 shows a novel extrinsic calibration method for a 3D LiDAR and a camera that resolves the drawbacks of existing methods. In place of the original MI, we employ BLSMI to robustly estimate statistical dependence between LiDAR point clouds and camera images. The experimental results demonstrate that our calibration method using BLSMI exhibits superior calibration accuracy compared to the existing methods, particularly under noisy outdoor environments.

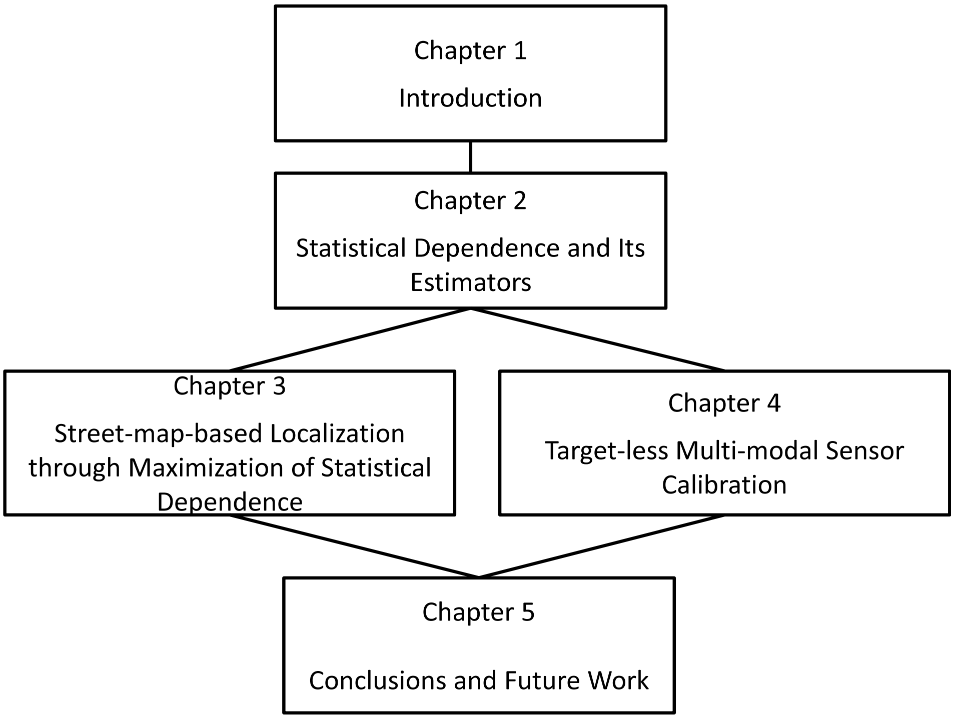


Figure 1.3: Organization of this thesis.

Finally, our conclusions along with a discussion of future research directions are given in Chapter 5.

Chapter 2

Statistical Dependence and Its Estimators

This chapter reviews statistical dependence which is the primary tool employed in the later chapters. In addition, an overview of a novel dependence estimator that we developed is presented along with other existing dependence estimators.

2.1 Mutual Information

Here, we review mutual information and its estimators.

2.1.1 Overview of Mutual Information

We consider two continuous random vectors $\boldsymbol{X} \in \mathbb{R}^{d_x}$ and $\boldsymbol{Y} \in \mathbb{R}^{d_y}$ that obey the joint probability density $p(\boldsymbol{x}, \boldsymbol{y})$. By the definition of statistical independence, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are statistically independent if and only if

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}),$$

where

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{y}, \quad p(\boldsymbol{y}) = \int p(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{x}.$$

Therefore, we can consider that $\boldsymbol{X}$ and $\boldsymbol{Y}$ have some dependence if the joint density $p(\boldsymbol{x}, \boldsymbol{y})$ and the product of two densities $p(\boldsymbol{x})p(\boldsymbol{y})$ are different. The dependence can be measured by "distance" between these two components. For example, the definition of mutual information (MI) [Shannon, 1948], which is one of the most popular dependence measures, is given as follows:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) := \iint p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}. \tag{2.1}$$

MI can be interpreted as the Kullback-Leibler divergence between $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$. MI is always non-negative and satisfies

$$\mathrm{MI} = 0 \iff p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y}).$$

MI has a variety of applications including independence test, feature selection and image registration [Pethel and Hahs, 2014] [Peng *et al.*, 2005] [Pluim *et al.*, 2003].

2.1.2 Mutual Information Estimators

There are various ways to estimate mutual information. Here we discuss several estimators that are commonly employed.

2.1.2.1 Discretized Mutual Information

One of the naïve approaches is to discretize continuous vectors and apply the discretized version of MI. MI for discretized random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined as follows:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{\boldsymbol{x}} \sum_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}. \tag{2.2}$$

We can easily calculate (2.2) by:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{H}(\boldsymbol{X}) + \mathrm{H}(\boldsymbol{Y}) - \mathrm{H}(\boldsymbol{X}, \boldsymbol{Y}), \tag{2.3}$$

where

$$\begin{aligned}\mathrm{H}(\boldsymbol{X}) &= -\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(\boldsymbol{x}),\\ \mathrm{H}(\boldsymbol{Y}) &= -\sum_{\boldsymbol{y}} p(\boldsymbol{y}) \log p(\boldsymbol{y}),\\ \mathrm{H}(\boldsymbol{X},\boldsymbol{Y}) &= -\sum_{\boldsymbol{x}}\sum_{\boldsymbol{y}} p(\boldsymbol{x},\boldsymbol{y}) \log p(\boldsymbol{x},\boldsymbol{y}).\end{aligned}$$

H is known as *Shannon's entropy* [Shannon, 1948].

A simple way to apply this method to continuous random vectors is to partition each domain of random vectors in to a finite set. Then, calculate $p(\boldsymbol{x})$, $p(\boldsymbol{y})$, and $p(\boldsymbol{x},\boldsymbol{y})$ by constructing the histograms. The problem of this method is, however, that the estimation performance depends on the discretization. In addition, it is not practical to apply this method to high-dimensional random vectors. In high-dimensional setup, both memory and samples can easily short because the total number of bins increases exponentially as the growth of the dimensionality [Scott and Sain, 2005]. The sample shortage means that the most of the bins result in zero count.

2.1.2.2 Use of Kernel Density Estimation

Another naïve approach is using kernel density estimation (KDE) [Scott, 1992] to estimate $p(\boldsymbol{x})$, $p(\boldsymbol{y})$, and $p(\boldsymbol{x},\boldsymbol{y})$ and substitute them into the original equation (2.1). KDE estimates a density function $f(\boldsymbol{x})$ using its i.i.d. samples $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n$ by

$$\widehat{f}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\boldsymbol{x},\tilde{\boldsymbol{x}}_i),$$

where K is a *kernel* function. A typical choice for the kernel is the Gaussian kernel defined as:

$$K(\boldsymbol{x},\tilde{\boldsymbol{x}}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{||\boldsymbol{x}-\tilde{\boldsymbol{x}}||^2}{2\sigma^2}\right),$$

where σ is the bandwidth.

This approach typically gives a better estimation result than the discretized one [Moon *et al.*, 1995]. However, this method can result in large estimation errors because the error in estimating individual components $p(\boldsymbol{x})$, $p(\boldsymbol{y})$, and $p(\boldsymbol{x}, \boldsymbol{y})$ can be combined and magnified.

2.1.2.3 Maximum Likelihood Mutual Information

Recently, a method called Maximum Likelihood Mutual Information (MLMI) that solves the above-mentioned problem was proposed [Suzuki *et al.*, 2009b]. Instead of individually estimating each density, MLMI models the density ratio,

$$r(\boldsymbol{x}, \boldsymbol{y}) := \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})},$$

and estimates it directly. MLMI is known to be superior to the KDE-based method in terms of estimation error [Suzuki *et al.*, 2009b].

2.1.3 Disadvantages of Mutual Information

Thus far, we reviewed mutual information and its estimators. The discretized MI estimator is easy to compute; however the discretization causes estimation errors. The problem can be avoided by using the KDE-based estimator, which estimates MI in the continuous space. While the KDE-based estimator involves the estimation of densities $p(\boldsymbol{x})$, $p(\boldsymbol{y})$, and $p(\boldsymbol{x}, \boldsymbol{y})$, combining the estimate components can cause magnified estimation errors. MLMI was developed to solve the problem by directly estimating the density ratio $p(\boldsymbol{x}, \boldsymbol{y})/p(\boldsymbol{x})p(\boldsymbol{y})$.

Nevertheless, there remain several problems of MI. The estimation of MI is noise-sensitive because of the logarithm in the equation (2.1). Also, the logarithm renders the estimators computationally rather expensive. In the following sections, we discuss several alternatives of MI.

2.2 Squared-loss Mutual Information

Recently, a variant of MI called squared-loss mutual information (SMI) [Suzuki *et al.*, 2009a] which has several advantages over MI is also employed. Here we introduce SMI and a computationally efficient SMI estimator.

2.2.1 Overview of Squared-loss Mutual Information

The definition of SMI is given as follows:

$$\mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{2} \iint p(\boldsymbol{x})p(\boldsymbol{y}) \left(\frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}. \tag{2.4}$$

While MI can be regarded as the Kullback-Leibler divergence, SMI corresponds to the Pearson divergence between $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$ [Suzuki *et al.*, 2009a]. Because both the Kullback-Leibler divergence and Pearson divergence are of the f-divergence family, they share similar properties. However, as can be seen from (2.4), SMI does not contain logarithm; therefore, SMI is robust to outliers compared to MI. Recently, a computationally efficient estimator of SMI has been developed [Sakai and Sugiyama, 2014]. Because of the advantages above, we use SMI in Chapters 4 and 3.

2.2.2 Least-squares Mutual Information

Here we review LSMI, a computationally efficient estimator of SMI developed by Sakai and Sugiyama [Sakai and Sugiyama, 2014].

2.2.2.1 The LSMI Algorithm

In LSMI, the density ratio,

$$r(\boldsymbol{x}, \boldsymbol{y}) := \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}, \tag{2.5}$$

is estimated using the following multiplicative kernel model:

$$r_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{y}) := \sum_{l=1}^{b} \sum_{l'=1}^{b} \alpha_{l,l'} G(\boldsymbol{x}, \tilde{\boldsymbol{x}}_l) L(\boldsymbol{y}, \tilde{\boldsymbol{y}}_{l'}).$$

Here, G and L are kernel basis functions. We employ the Gaussian kernel functions:

$$G(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \exp\left(-\frac{||\boldsymbol{x} - \tilde{\boldsymbol{x}}||^2}{2\sigma^2} \right),$$
$$L(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \exp\left(-\frac{||\boldsymbol{y} - \tilde{\boldsymbol{y}}||^2}{2\sigma^2} \right),$$

where $\sigma > 0$ is the Gaussian bandwidth. Kernel centers $\{(\tilde{\boldsymbol{x}}_l, \tilde{\boldsymbol{y}}_l)\}_{l=1}^{b}$ are randomly chosen from the input data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$. The parameter $\boldsymbol{\alpha}$ ($b \times b$ matrix) is estimated to minimize the following squared loss:

$$J'(\boldsymbol{\alpha}) = \iint (r_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{y}) - r(\boldsymbol{x}, \boldsymbol{y}))^2 p(\boldsymbol{x}) p(\boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y}. \tag{2.6}$$

To avoid overfitting, the ℓ_2 regularization term is added to the squared-loss and the following objective function is obtained:

$$J(\boldsymbol{\alpha}) := \iint (r_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{y}) - r(\boldsymbol{x}, \boldsymbol{y}))^2 p(\boldsymbol{x}) p(\boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y} + \lambda ||\boldsymbol{\alpha}||_2,$$

where $\lambda > 0$ is the regularization parameter. The expectations are approximated by the empirical averages and the derivative of $J(\boldsymbol{\alpha})$ is set to zero. Then, the minimizer of J can be obtained by solving the following discrete-time Sylvester equation about $\boldsymbol{\alpha}$:

$$\frac{1}{n^2} \boldsymbol{G}^T \boldsymbol{G} \boldsymbol{\alpha} \boldsymbol{L}^T \boldsymbol{L} + \lambda \boldsymbol{\alpha} = \frac{1}{n} \boldsymbol{G}^T \boldsymbol{L}. \tag{2.7}$$

The elements of $\boldsymbol{G}$ and $\boldsymbol{L}$ (both are $n \times b$ matrices) are given by $G_{i,l} = G(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_l)$, $L_{i,l} = L(\boldsymbol{y}_i, \tilde{\boldsymbol{y}}_l)$. Finally, estimated SMI is obtained using $\widehat{\boldsymbol{\alpha}}$, which is the solution of (2.7), as follows:

$$\mathrm{LSMI} = \frac{1}{2n} \operatorname{tr}(\boldsymbol{G} \widehat{\boldsymbol{\alpha}} \boldsymbol{L}^T) - \frac{1}{2}. \tag{2.8}$$

The Gaussian bandwidth σ and the regularization parameter λ are hyper-parameters which affect the performance of LSMI. In practice, it is important to choose appropriate hyper-parameters.

2.2.2.2 Hyper-parameter Selection in LSMI

In LSMI, hyper-parameters can be determined by K-fold cross validation with respect to the criterion J' defined by 2.6. The procedure of the cross validation is as follows. The input data $\mathcal{D}$ is split into K disjoint sets $\{\mathcal{D}_k\}_{k=1}^{K}$. The size of each set is approximately equalized. For each subset $\mathcal{D}_k$, the hold-out error,

$$J_k^{\mathrm{CV}} = \frac{1}{2|\mathcal{D}_k|^2} \sum_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_k} \widehat{r}_k(\boldsymbol{x}, \boldsymbol{y})^2 - \frac{1}{|\mathcal{D}_k|} \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_k} \widehat{r}_k(\boldsymbol{x}, \boldsymbol{y}),$$

is calculated, where $\widehat{r}_k(\boldsymbol{x}, \boldsymbol{y})$ is the density ratio function (2.5) estimated using $\mathcal{D} \backslash \mathcal{D}_k$ (the input data excluding $\mathcal{D}_k$). Finally, hyper-parameters that minimize the average error,

$$J^{\mathrm{CV}} = \frac{1}{K} \sum_{k=1}^{K} J_k^{\mathrm{CV}},$$

are chosen from several candidates.

2.2.3 Bagged LSMI

During the study of calibration (Chapter 4), we observed that the gradient-based maximization of LSMI sometimes got trapped in local optima. We determined that the problem is caused by the variance of LSMI estimation which renders the calculation of the numerical gradient inaccurate. To overcome the difficulty, we developed a novel dependence estimator called *bagged LSMI (BLSMI)* that incorporates the bagging technique [Breiman, 1996]. Bagging reduces the estimation variance by averaging multiple solutions obtained from resampled data sets.

The estimation procedure of BLSMI is as follows. We generate bootstrap data sets $\{\mathcal{D}_m\}_{m=1}^{M}$ by random subsampling from the original input data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$. We estimate SMI for each bootstrap data set and calculate the mean of the estimated values as

$$\mathrm{BLSMI}(\mathcal{D}) = \frac{1}{M} \sum_{m=1}^{M} \mathrm{LSMI}(\mathcal{D}_m). \tag{2.9}$$

The effectiveness of BLSMI is evaluated in experiments presented in Chapter 4.

2.3 Other Dependence Measures

There are a large number of dependence measures other than MI and SMI; here we introduce several of them that are employed in the experiments shown later.

2.3.1 Quadratic Mutual Information

Quadratic mutual information [Torkkola, 2003] is defined as follows:

$$\mathrm{QMI}(\boldsymbol{X}, \boldsymbol{Y}) := \iint \left(p(\boldsymbol{x}, \boldsymbol{y}) - p(\boldsymbol{x})p(\boldsymbol{y})\right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}.$$

QMI can be interpreted as the L2 distance between $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$. Similarly to SMI, QMI does not contain logarithm; therefore SMI is also considered as robust dependence measure [Sainui and Sugiyama, 2013]. Recently, an estimator of QMI called *least-squares quadratic mutual information (LSQMI)*, has been proposed and successfully applied to dimension reduction [Sainui and Sugiyama, 2013].

2.3.2 Conditional Entropy

Conditional entropy (CE) can also be used to measure dependence. The conditional entropy of $\boldsymbol{Y}$ given $\boldsymbol{X}$ is defined as:

$$\mathrm{CE}(\boldsymbol{Y}|\boldsymbol{X}) := -\iint p(\boldsymbol{x}, \boldsymbol{y}) \log p(\boldsymbol{y}|\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}.$$

$\mathrm{CE}(\boldsymbol{Y}|\boldsymbol{X})$ can be interpreted as the negative Kullback-Leibler divergence between $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x})$.

The relationship between MI and CE can be observed from the following equation.

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) = H(\boldsymbol{Y}) - \mathrm{CE}(\boldsymbol{Y}|\boldsymbol{X}). \tag{2.10}$$

This means that increasing MI corresponds to decreasing CE. Unlike MI, CE measures "directed" dependence; this means that $\mathrm{CE}(\boldsymbol{Y}|\boldsymbol{X}) = \mathrm{CE}(\boldsymbol{X}|\boldsymbol{Y})$ does not generally hold.

Similarly to the discussion of MI and SMI, we can consider the Pearson-divergence version of conditional entropy:

$$\mathrm{SCE}(\boldsymbol{Y}|\boldsymbol{X}) := -\frac{1}{2} \iint p(\boldsymbol{x}) \left(p(\boldsymbol{y}|\boldsymbol{x}) - 1\right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y},$$

which is known as *squared-loss conditional entropy (SCE)* [Tangkaratt *et al.*, 2015]. A method that estimates SCE using the framework of the density ratio estimation has also been developed [Tangkaratt *et al.*, 2015].

2.3.3 Distance Correlation

Recently, a novel dependence measure called the *distance correlation (dCor)* has been proposed [Székely *et al.*, 2007]. dCor can be considered as the generalization of the Pearson correlation [Székely and Rizzo, 2009] for non-linear dependence. dCor is defined as follows:

$$\mathrm{dCor}(\boldsymbol{X}, \boldsymbol{Y}) := \begin{cases} 0 & \text{if } \mathrm{dCov}(\boldsymbol{X}, \boldsymbol{Y}) = 0, \\ \dfrac{\mathrm{dCov}(\boldsymbol{X}, \boldsymbol{Y})}{\mathrm{dCov}(\boldsymbol{X}, \boldsymbol{X})\mathrm{dCov}(\boldsymbol{Y}, \boldsymbol{Y})} & \text{otherwise,} \end{cases}$$

where dCov is the *distance covariance* [Székely *et al.*, 2007] defined as

$$\mathrm{dCov}(\boldsymbol{X}, \boldsymbol{Y}) := ||\phi_{\boldsymbol{X},\boldsymbol{Y}}(s, t) - \phi_{\boldsymbol{X}}(s)\phi_{\boldsymbol{Y}}(t)||_2.$$

Here, $\phi_{\boldsymbol{X},\boldsymbol{Y}}(s, t)$ is the joint characteristic function of $\boldsymbol{X}$ and $\boldsymbol{Y}$, and $\phi_{\boldsymbol{X}}(s)$ and $\phi_{\boldsymbol{Y}}(t)$ are the marginal characteristic functions.

One of the advantages of dCor is that it can be obtained by simple calculation using the pairwise distance between samples [Székely *et al.*, 2007].

Chapter 3

Street-map-based Localization through Maximization of Statistical Dependence

3.1 Introduction

This chapter considers the problem of mobile robot localization. Recently, localization methods based on detailed maps constructed using simultaneous localization and mapping (SLAM) have been widely used with great success [Levinson and Thrun, 2010] [Kümmerle *et al.*, 2013] [Joshi and James, 2015]. However, the construction of such maps is very costly, because the necessary data collection involves a manual procedure (e.g., manual robot navigation). This cost can be eliminated by reusing existing street maps. Localization using street maps has primarily been studied with regard to roadways [Brakatsoulas *et al.*, 2005] [Chausse *et al.*, 2005], but no methods that are effective for sidewalks have been reported. One of the difficulties with regard to sidewalks is that road boundaries are not clearly marked; road boundary detection can be even more difficult than localization. However, localization is possible without road boundary detection. Here, we present a novel localization approach referred to as *dependence maximization localization*, which was inspired by multi-modal image registration.

The method presented in this chapter localizes a robot by maximizing the latent dependence between sensor data and a street map. Squared-loss mutual information (SMI), a variant of mutual information (MI) is employed to measure this dependence.

3.1.1 Related Work

Existing street-map-based localization methods can be classified into two categories. Methods designed to assist human car drivers and those for autonomous navigation. Typical applications within the first category are car navigation systems and advanced driver assistance systems (ADAS). A typical localization approach employed in such cases is to match a vehicle's trajectory with road network information extracted from street maps, assuming that the vehicle is always on the road. Methods using global navigation systems (GPS) for trajectory estimation have been studied for over a decade [Brakatsoulas *et al.*, 2005]. Recently, several authors have proposed the use of visual odometry, and global localization without GPS has been shown to be possible [Parra Alonso *et al.*, 2012] [Brubaker *et al.*, 2013] [Floros *et al.*, 2013]. However, the limitation of this approach is that the precise position with respect to the road, which is crucial information for autonomous navigation, cannot be estimated.

Although localization using street maps for autonomous navigation has also been studied, the majority of such studies have focused on roadways. As roadway boundaries are typically clearly marked, a large number of methods involve matching of the lane marking to the road boundaries on the map [Chausse *et al.*, 2005] [Tao *et al.*, 2013] [Lu *et al.*, 2014]. Highly accurate localization results have been reported in such cases. However, several authors have proposed methods that do not use lane markings. For example, the method proposed by Morales *et al.* estimates the offset between the road center and the robot from LiDAR measurements [Morales *et al.*, 2009]. In that method, the road center is estimated from LiDAR measurements with the assumption that the road boundaries can be determined from the height information. However, sidewalk environments vary considerably and, in some cases, road boundaries cannot be estimated from height information alone. Therefore, the above method is not directly applicable to sidewalks. In contrast, Hentschel and Wagner have proposed a technique through which building boundaries are detected and correlated with the map [Hentschel and Wagner, 2010]. The advantage of using building boundaries is that they can be easily detected from LiDAR measurements. However, this method is only applicable in environments where buildings are sufficiently observable and correctly mapped.

The difficulty of localization in sidewalks has been discussed by Irie and Tomono [Irie and Tomono, 2012]. They have proposed the application of object recognition to manage varying road boundaries along sidewalks. In this approach, the recognized objects are matched with object prior probabilities embedded in the street map (e.g., the likelihood of curbs is high at the boundary between the sidewalk and roadway). Although this proposed framework is general, object prior probabilities vary from place to place. Therefore, applying their method to *new* environments should involve additional costs to determine the appropriate object prior probabilities.

3.1.2 Contribution

In this chapter we propose a novel method that relaxes the limitations imposed on previous techniques. That is, our method does not explicitly detect road boundaries or recognize objects. Instead, sensor observations are directly matched to a map using a variant of MI. Although MI has already been used to register multi-modal sensor data [Pluim *et al.*, 2003] [Alempijevic *et al.*, 2009] [Pandey *et al.*, 2014], to the best of our knowledge, it has never been applied to street-map-based localization. The advantages of our method compared to existing methods can be summarized as follows:

- Robot trajectories are not limited within the road network.
- Detection of road or building boundaries is not required.
- No prior information e.g., on the colors, textures, or heights of objects in the environment is required.
- No other information besides boundary lines is required from the map. This indicates that the efficacy of our method is independent of whether or not the regions on the map correspond to roads, buildings, etc.

Our method is the first to possess all the above characteristics.

In addition to outlining the proposed method, we also describe certain implementation details that render our method computationally efficient for practical navigation use. We evaluate the performance of our method via extensive experiments using real-world datasets, and we empirically demonstrate that SMI exhibits superior functionality among several dependence measures.

Successful results obtained for long distance (up to 4.5 km) position tracking experiments support the efficacy of the proposed method.

The main contributions of the chapter can be summarized as follows:

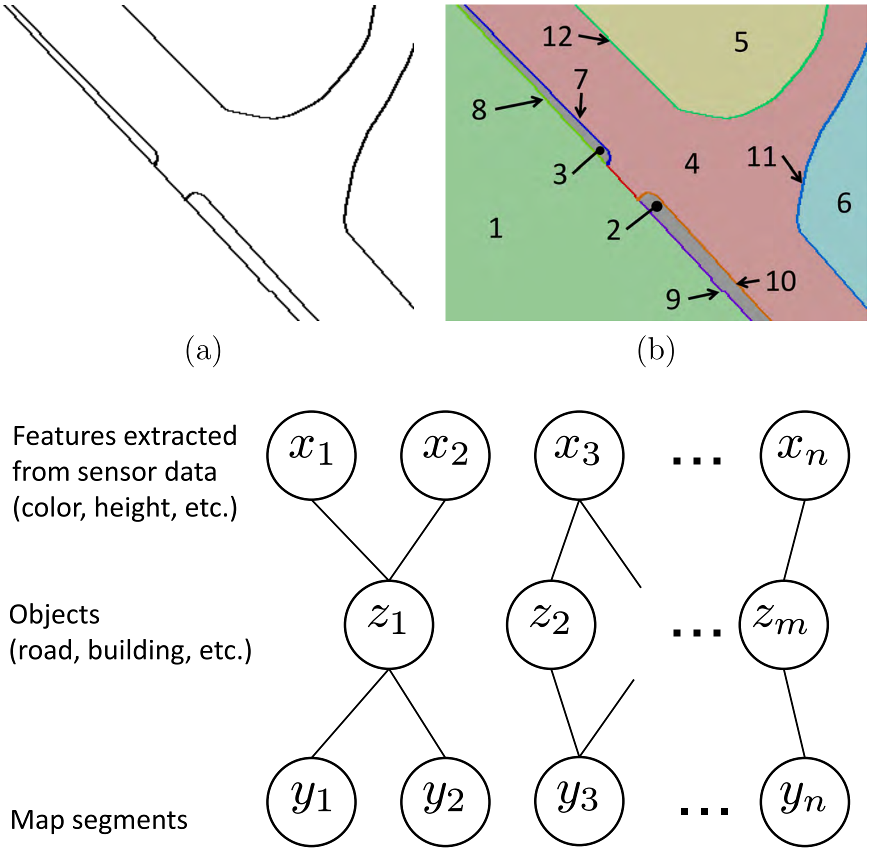
- Introduction of the dependence maximization approach to street-map-based localization.
- Computationally efficient implementation of the proposed method.
- The finding that SMI has superior functionality among several different dependence measures.
- Empirical demonstrations using real-world datasets showing that our method is both effective and sufficiently computationally efficient for practical use.

3.2 Proposed Method

As we employ a 2D street map, our method estimates the 2D pose (position and orientation) of the robot $\boldsymbol{w} = (w_x, w_y, w_\theta)$. We are particularly interested in localization for navigation; we focus on position tracking under the assumption that the initial position of the robot is known (with some errors).

3.2.1 Map Representation

Our method uses 2D street maps consisting of boundary lines as shown in Figure 3.1 (a). During pre-processing, each closed region and boundary line on the map is given a unique ID. We refer to the IDs as *segment IDs* in this paper. Sample segment IDs are shown in Figure 3.1 (b). The assumption behind this procedure is that map segments are defined according to the objects in the environment. In other words, we expect that similar objects exist within a segment and different objects exist in different segments (e.g., ID = 1 corresponds to buildings and ID = 7 corresponds to curbs). Because sensor observations naturally depend on the *objects* being observed, sensor data and segment IDs should have a latent dependence. An example of the latent dependence between sensor data, objects, and the map is depicted in Figure 3.1 (c). The proposed method exploits this dependence for robot localization.



(a) (b)

(c)

Figure 3.1: (a) Sample map used in proposed method. (b) Map interpretation. Different colors indicate different segment IDs. (c) Illustrative model of the world. The map and sensor data have latent dependence.

3.2.2 Approach

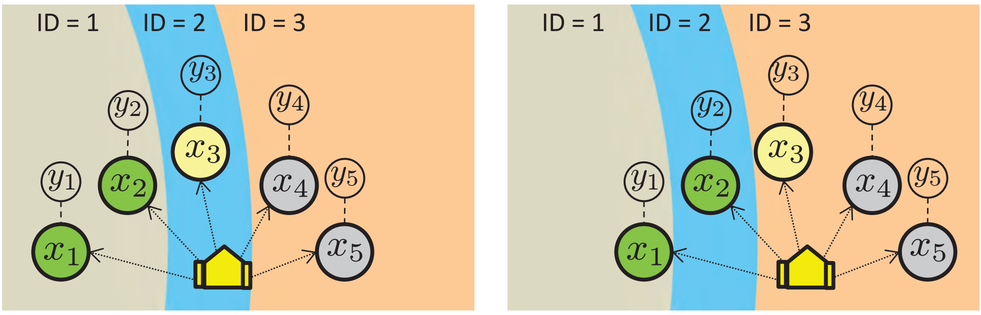
Our approach was inspired by a method used for image registration problems. One of the well-known image registration methods is maximization of the normalized cross correlation (NCC). Maximizing the NCC can be interpreted as maximizing the dependence between the pixel intensities of two images. It is assumed that the same object exhibits similar intensity values in two images; therefore the pixel intensities of these two images should have *linear* dependence. However, if two images are of different modalities (e.g., with regard to the registration of medical computed tomography (CT) and magnetic resonance (MR) images [Collignon *et al.*, 1995]) the pixel intensities can have *non-linear* dependence. That is, some objects rendered in white in CT images appear black in MR images, and others do not. Maximizing MI is known to be effective for this kind of multi-modal image registration [Pluim *et al.*, 2003] because MI can measure non-linear dependence. Our method applies this concept to robot localization; the sensor data and map segments may have non-linear dependence and, thus we maximize the MI-based dependence measure to localize a robot.

3.2.3 Dependence Maximization Localization

The robot observes n different positions using onboard sensors and features are extracted from the sensor data (such as colors and shapes). We denote the extracted features by $\{\boldsymbol{x}_i\}_{i=1}^n$. Using a robot pose hypothesis $\boldsymbol{w}$, we find the segment IDs that correspond to the segments in which the features are located; these are denoted by $\{y_i^{(\boldsymbol{w})}\}_{i=1}^n$ (it is assumed that the relative positions between the robot and the sensor data source are known). Consequently, we obtain pairs of feature and segment ID $\{(\boldsymbol{x}_i, y_i^{(\boldsymbol{w})})\}_{i=1}^n$, which are used as input data.

The features and map segment IDs are denoted by random variables $\boldsymbol{X}$ and $\boldsymbol{Y}^{(\boldsymbol{w})}$. By assuming that the input data is randomly generated from a joint probability distribution $p(\boldsymbol{X}, \boldsymbol{Y}^{(\boldsymbol{w})})$, we search for the hypothesis $\boldsymbol{w}$ that maximizes a dependence measure. The estimated robot pose $\hat{\boldsymbol{w}}$ can be obtained by

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}}\, f(\boldsymbol{X}, \boldsymbol{Y}^{(\boldsymbol{w})}). \tag{3.1}$$



(a) correct localization
(high dependence)

(b) incorrect localization
(low dependence)

FIGURE 3.2: Illustration of our localization approach. In this example, the robot obtains five color features ($\boldsymbol{x}_1, ..., \boldsymbol{x}_5$) from different positions. Corresponding map segment IDs ($y_1, ..., y_5$) are determined using the current robot pose hypothesis. (a) The dependence between features $\boldsymbol{x}$ and segment IDs y is high when the robot is correctly localized because y can be uniquely determined from $\boldsymbol{x}$. (b) The dependence decreases when the robot is incorrectly localized.

We employ SMI [Suzuki *et al.*, 2009a] as the measure function f. As already discussed in Section 2.2, SMI is robust against outliers and can be estimated efficiently compared to ordinary MI.

To facilitate an intuitive understanding of our method, an illustrative example is provided in Figure 3.2. In this example, the robot obtains five color features ($\boldsymbol{x}_1, ..., \boldsymbol{x}_5$), where $\boldsymbol{x}_1 = \boldsymbol{x}_2 =$ green, $\boldsymbol{x}_3 =$ yellow and $\boldsymbol{x}_4 = \boldsymbol{x}_5 =$ gray. When the robot position is correct (Figure 3.2 (a)), the corresponding segment IDs are $y_1 = y_2 = 1$, $y_3 = 2$, $y_4 = y_5 = 3$. The conditional entropy of the features given map segment is small, because each segment of the map corresponds to a single color. Therefore, the dependence between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is high. On the other hand, when the robot position is incorrect as shown in Figure 3.2 (b), segment ID $= 3$ corresponds to multiple colors, yielding a reduced dependence.

3.3 Position Tracking Using a Particle Filter

Above, we have discussed robot localization from one-shot observation. Now, we move to time-series localization of a moving robot. This problem is also known as position tracking and has been intensively studied for decades [Leonard and Durrant-Whyte, 1991] [Thrun *et al.*, 2000]. At present, methods fusing a robot's motion estimation and observations using a particle filter are widely used [Kümmerle *et al.*, 2013] [Chausse *et al.*, 2005] [Hentschel and Wagner, 2010] . In this section, we describe the integration of dependence maximization localization with a particle filter.

3.3.1 SMI-based Particle Filter Localization

In each particle-filter prediction step, the robot predicts its pose using the motion $\boldsymbol{u}_t$, which denotes the difference in the estimated pose between the previous time step $t-1$ and the current step t; this information can be obtained through odometry, for example. The robot pose for the j-th particle is denoted by $\boldsymbol{w}_j$ and the particles are drawn from a proposal distribution

$$\boldsymbol{w}_j \sim p(\boldsymbol{w}^t | \boldsymbol{w}_j^{t-1}, \boldsymbol{u}_t). \tag{3.2}$$

During the particle-filter update steps, the robot extracts features $\{\boldsymbol{x}_i\}_{i=1}^n$ from the collected sensor data. The particles are then resampled according to the weight (W) proportional to the observation likelihood as

$$W_j \propto p(\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n | \boldsymbol{w}_j, \boldsymbol{M}). \tag{3.3}$$

Here, the map is denoted by $\boldsymbol{M}$. We estimate the SMI for each particle and interpret the result as the observation likelihood

$$p(\boldsymbol{X} | \boldsymbol{w}_j, \boldsymbol{M}) = \mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y}^{(\boldsymbol{w}_j)}). \tag{3.4}$$

3.3.2 Computationally Efficient Implementation

The processing time of the update steps is another challenge for applying this method to online navigation. Although the LSMI core calculation is computationally efficient, the cross-validation process consumes a considerable amount of time. This can be a hindrance to the implementation of a real-time system. In our implementation, we reduce the number of cross-validation procedures under the assumption that the changes in the hyper parameters between successive update steps are small. More specifically, we augment the

Table 3.1: List of datasets used in experiments

Dataset	Description
A	37 monocular camera images (320×240) with manually labeled positions
B	575 monocular camera images (320×200) with GPS reference positions
C	3D LiDAR scans and omnidirectional camera images, and 150m odometry with GPS reference trajectory
D	3D LiDAR scans and omnidirectional camera images, and 4.5km odometry with GPS reference trajectory

state vector $\boldsymbol{s}$ by appending σ and λ such that

$$\boldsymbol{s}_j = (\boldsymbol{w}_j, \sigma_j, \lambda_j). \tag{3.5}$$

In the very first update step, we perform cross-validation for all particles and store the chosen parameters in $\boldsymbol{s}$. In the subsequent update, we define an *omission rate* and randomly choose particles for which the cross-validation step is omitted. For example, if the omission rate is set to 70%, we perform cross-validation for only 30% of the particles and reuse the previously chosen hyper parameters for the remaining particles. In this manner, we can reduce the processing time by increasing the omission rate.

3.4 Experiments

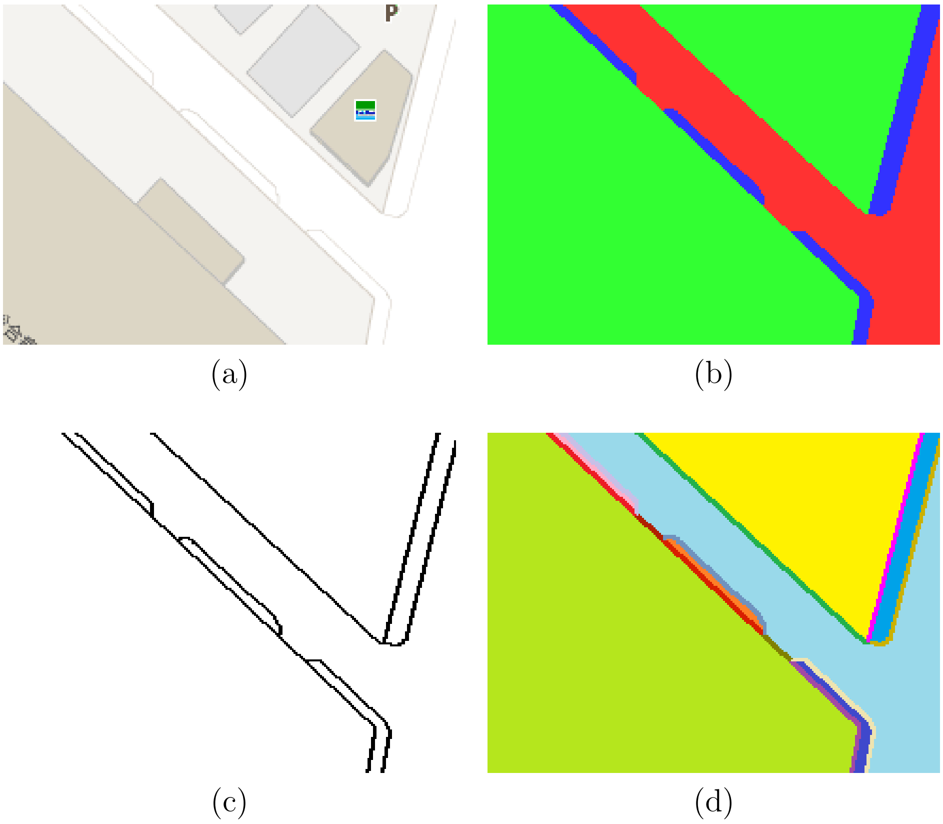
In this section we demonstrate the performance of our method through two types of experiments: localization from a single image and position tracking. We employ four datasets comprised of sensor data collected in an urban area of Narashino, Chiba, Japan (Table 3.1).

3.4.1 Embedding Segment IDs on Street Map

In the following experiments, we employed a street map generated by extracting segments in Google Maps™ [1]. A Google Maps™ screen capture was saved

---

[1] Although other maps such as OpenStreetMap [Haklay and Weber, 2008] can also be used, geometrically accurate maps are preferred. We employed Google Maps™ in this paper because, for our target area, it provided significantly more accurate sidewalk information than OpenStreetMap.



(a)
(b)
(c)
(d)

Figure 3.3: Illustration of map generation procedure. (a) Original map image captured from Google Maps$^{\mathrm{TM}}$. (b) Closed regions extracted manually. (c) Boundary lines detected using edge-detection algorithm. (d) Unique IDs assigned using flood-fill algorithm [Shaw, 2004]. Different colors indicate different IDs.

as an image and closed regions were extracted manually. Then, boundary lines were extracted using the Sobel filter. Finally, unique IDs were assigned to the segments using the flood-fill algorithm [Shaw, 2004]. Figure 3.3 illustrates the procedure[2].

3.4.2 Single Image 2-degrees-of-freedom Localization

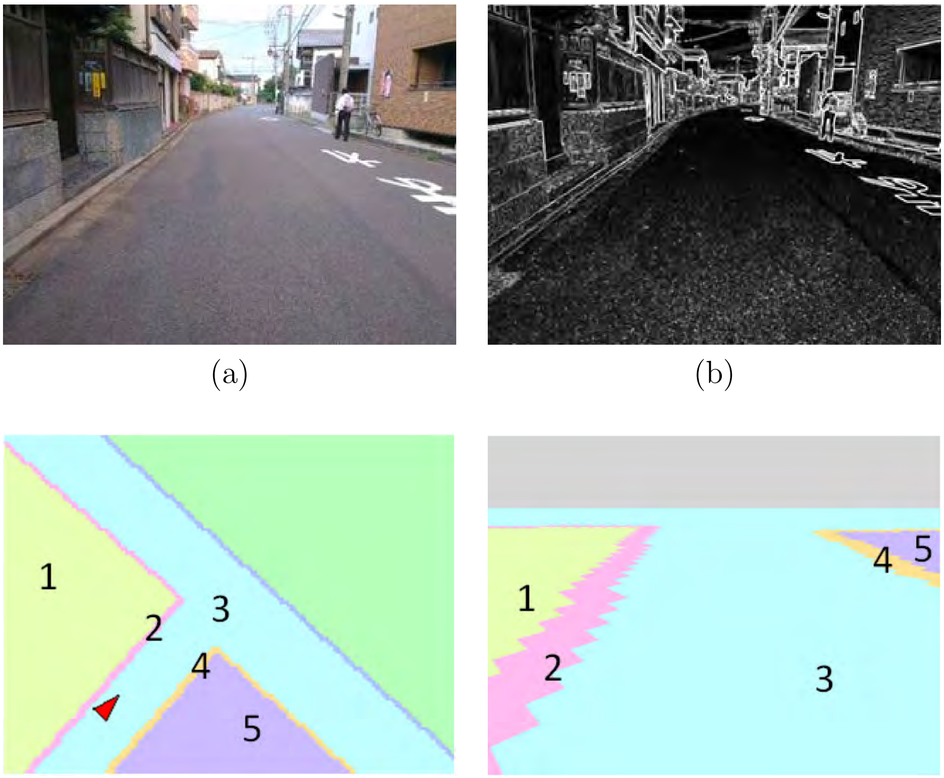
First, we evaluated the efficacy of SMI as a localization measure.

3.4.2.1 Setup

In the following experiments, 2-degree-of-freedom (DoF) robot poses (lateral position and orientation with respect to road) were estimated from single

---

[2] This process is unnecessary if we have vector data for the map.



(a) (b)

(c) (d)

FIGURE 3.4: Single image observation model. (a) Input image. (b) Extracted gradient feature. (c) Map and robot pose (red triangle). (d) Segment IDs projected onto image plane.

images. Note that 3-DoF localization was not performed, because 2D street maps are often non-discriminative in terms of the longitudinal direction and, also, the longitudinal position can be easily estimated from the robot's motion estimation (e.g., using odometry).

Colors and gradients were extracted from the images and used as features (summarized in Table 3.2). The set of features for the i-th pixel is denoted by $\boldsymbol{x}_i$. The correspondence between segment IDs and pixels was obtained by projecting the street map onto the image plane (Figure 3.4). The relative pose between the robot base and the camera was assumed to be known. The segment ID projected on the i-th pixel for $\boldsymbol{w}$ is denoted by $y_i^{(\boldsymbol{w})}$.

After the input data pairs $\{(\boldsymbol{x}_i, y_i^{(\boldsymbol{w})})\}_{i=1}^{n}$ were obtained, a grid search was used to find the position that maximizes Eq. (3.1). The search ranges were -2 m to $+2$ m with 0.2 m intervals for the lateral offset and $-20°$ to $+20°$ with 2° intervals for orientation.

3.4.2.2 Comparative Method Based on Road Boundary Matching

Although no existing localization method using the same setup is known, we implemented a comparative method based on road boundary matching. Road boundaries were detected using the road detection method proposed by Kong *et al.* [3] [Kong *et al.*, 2010], which uses vanishing point detection and does not require any prior information on the road color or texture. The detected road boundaries were matched with lines on the map. The robot poses at which the boundary lines match best were found using the same grid search mentioned above.

3.4.2.3 Results

We conducted single image localization experiments using two datasets: A) 37 images (32 from sidewalks and 5 from roadways) with manually given ground truth positions and, B) 575 images with GPS reference positions. We evaluated the accuracy, robustness, and processing time of our method. We also performed comparison of different dependence measures.

Localization Accuracy The localization results obtained for the two datasets are summarized in Figure 3.5. Our method outperformed the road matching method in terms of localization accuracy. Further, the average position and orientation errors were 0.45 m and 4.3° for dataset A and 0.64 m and 6.5° for dataset B. The map resolution was approximately 23.5 cm per pixel and, thus, a position error of 0.64 m corresponded to 2.7 pixels in the map image. We are of the view that the localization accuracy is reasonable, because the localization error includes errors of the map itself and the street map used here was not pixel-wise accurate.

Exemplary results for dataset A are shown in Figure 3.6 and 3.7, with LSMI plots against the position and orientation errors (the rightmost column). Successful examples for both methods are shown in Figure 3.6 (a) and (b). The LSMI plot for (a) exhibits a clear peak around the ground-truth position. Although the road detection method failed on Figure 3.6 (c-e), which was because of the confusing salient edges in the images, our method exhibited successful localization. Braille blocks caused several localization failures. Two

---

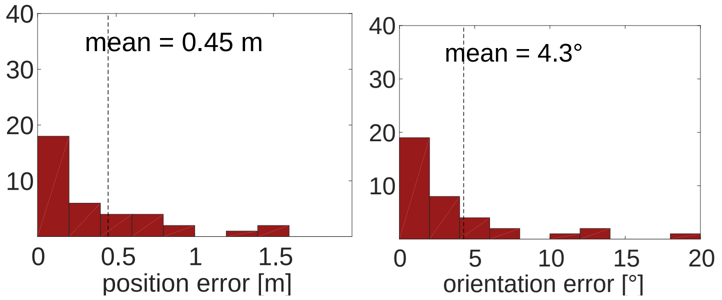
[3] We used the code downloaded from `http://web.mit.edu/huikong/www/code.html` .

Table 3.2: Features used in single image localization experiments.

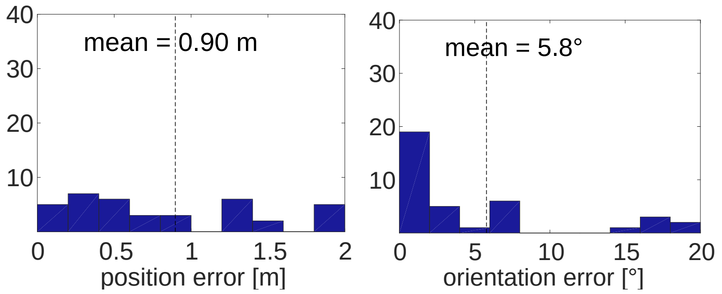
Category	Feature	Dimension
Color	R, G, B	3
Gradient	Sobel magnitude, orientation	2

peaks can be seen in the plot of Figure 3.7 (f); the highest peak appeared at the ground-truth and the second highest peak corresponded to the Braille blocks. Further, some of the localization errors were caused by map errors. An example of a road width error can be seen in Figure 3.7 (g). Both the road detection method and our method failed in the case shown in Figure 3.7 (h), where the road boundaries were unclear and the feature differences between the roadway, the sidewalk and the parking lot were very small. A challenging example of map errors is shown in Figure 3.7 (i). The left road boundary on the map appears curved, but this is inaccurate. Another challenge is coping with occlusions such as the example shown in Figure 3.7 (j); our method failed in this case because the blue tarp covering the road was of a similar color to the vehicle that appears on the right hand side of the image.

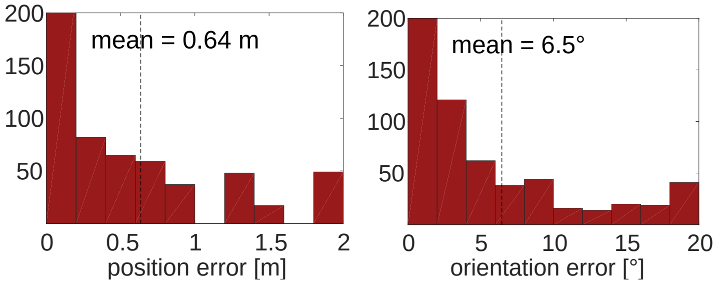
Exemplary results for dataset B are shown in Figure 3.8 and 3.9. As can be seen from the figures, the illumination conditions for dataset B was sometimes challenging. Localization errors caused by strong shadows would be the main reason of the performance difference between the two datasets.



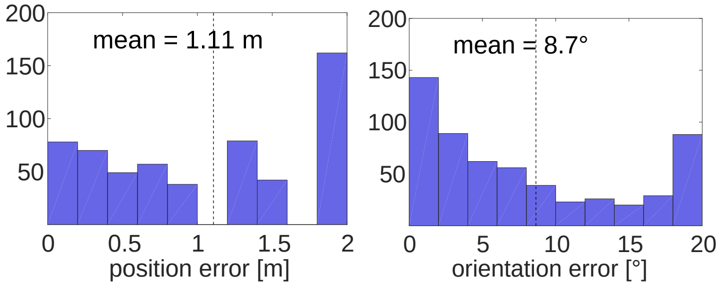
Dataset A – proposed method



Dataset A – road boundary matching



Dataset B – proposed method



Dataset B – road boundary matching

FIGURE 3.5: Histograms of position and orientation errors from single image localization. Top: Dataset A results (37 images). Bottom: Dataset B results (575 images).

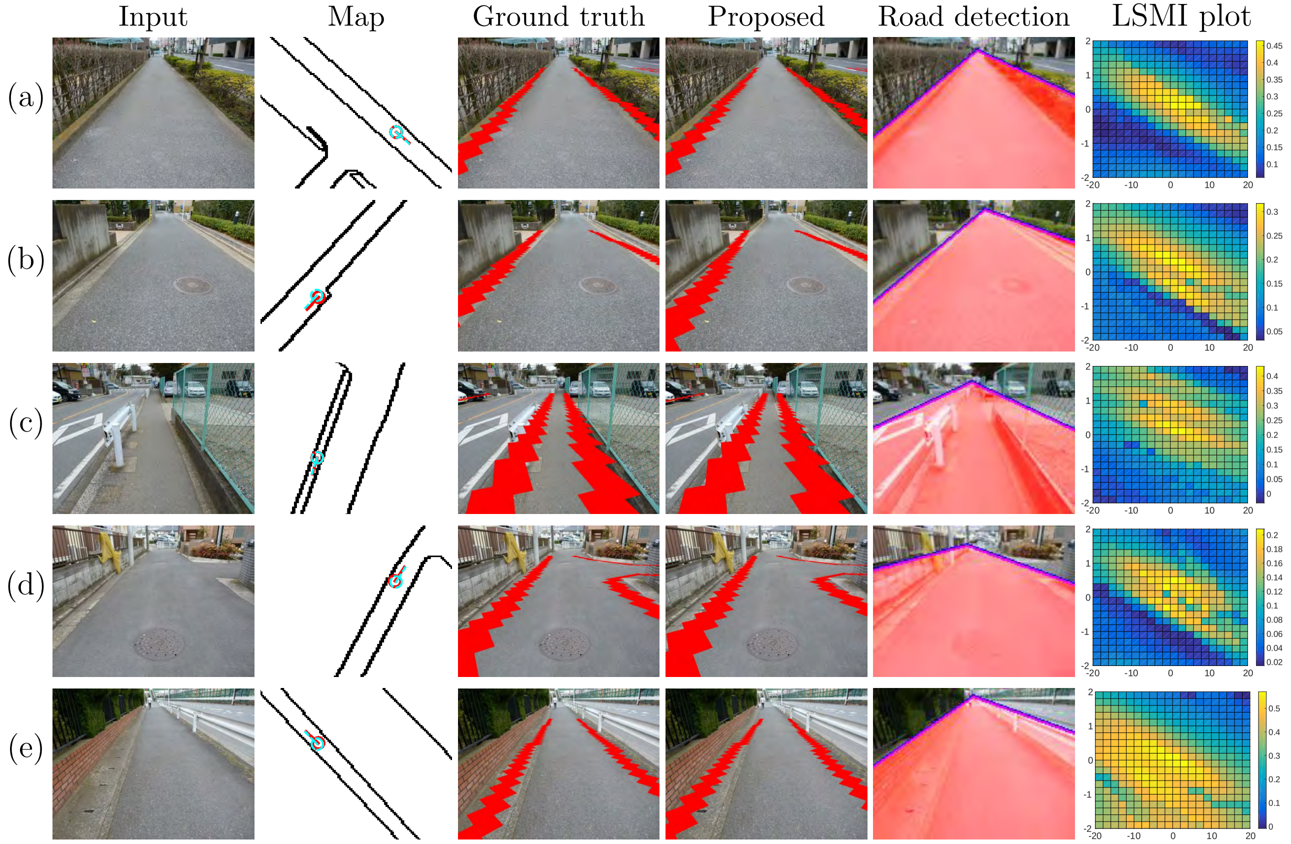


FIGURE 3.6: Exemplary results of localization experiments from dataset A. The localization results are shown by superimposing the road boundaries onto the images. In the maps the estimated pose and ground-truth positions are indicated in red and blue, respectively. The road detection results were obtained by the method in [Kong *et al.*, 2010]. The horizontal and vertical axes in the LSMI plots represent the orientation error [°] and the position error [m], respectively. In the comparative method, the detected road boundaries were matched with the map for localization.

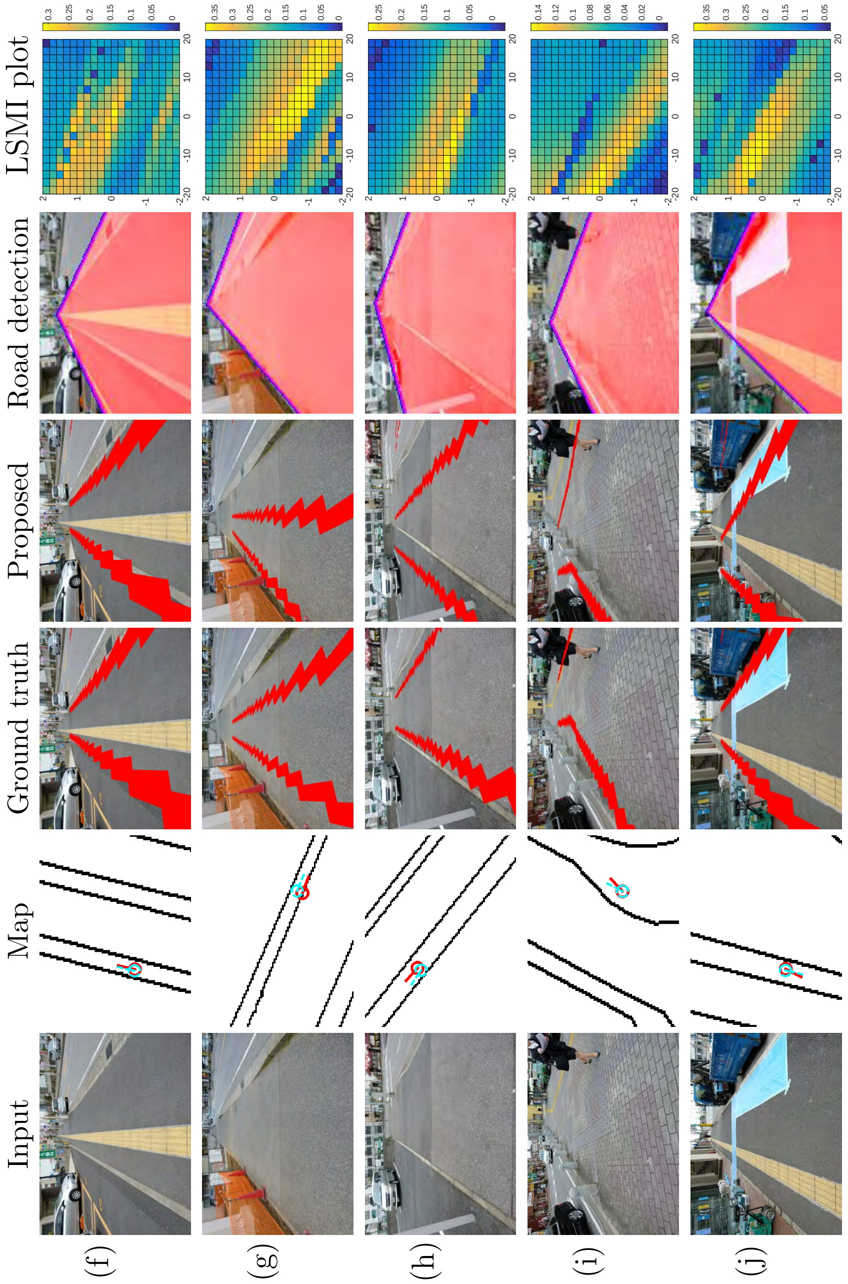


Figure 3.7: Challenging examples from dataset A.

Comparison of Dependence Measures We evaluated our proposed method using different dependence measures: ordinary MI, SMI, QMI, SCE, and dCor. Refer to Chapter 2 for details of these dependence measures. The results are summarized in Table 3.3 and Figure 3.10. Exemplary score distributions are shown in Figure 3.11. SMI performed best in terms of accuracy; both the position and orientation errors by SMI were significantly smaller ($p < 0.05$ by the paired t-test) than any other dependence measures. All dependence measures except MI are considered to be robust measures (MI's sensitivity can be seen from the spikes in the MI plot). QMI, which measures the dependence based on the ℓ_2 distance between $p(\boldsymbol{x}, y)$ and $p(\boldsymbol{x})p(y)$, exhibited the second best performance. SCE can also measure dependence robustly, and its score plot appears smooth; however, SCE($\boldsymbol{Y}|\boldsymbol{X}$) had high values where all observations indicated a single segment ID. Therefore the SCE performance was poor. In terms of the processing time, dCor exhibited shortest time and SMI performed the second best.

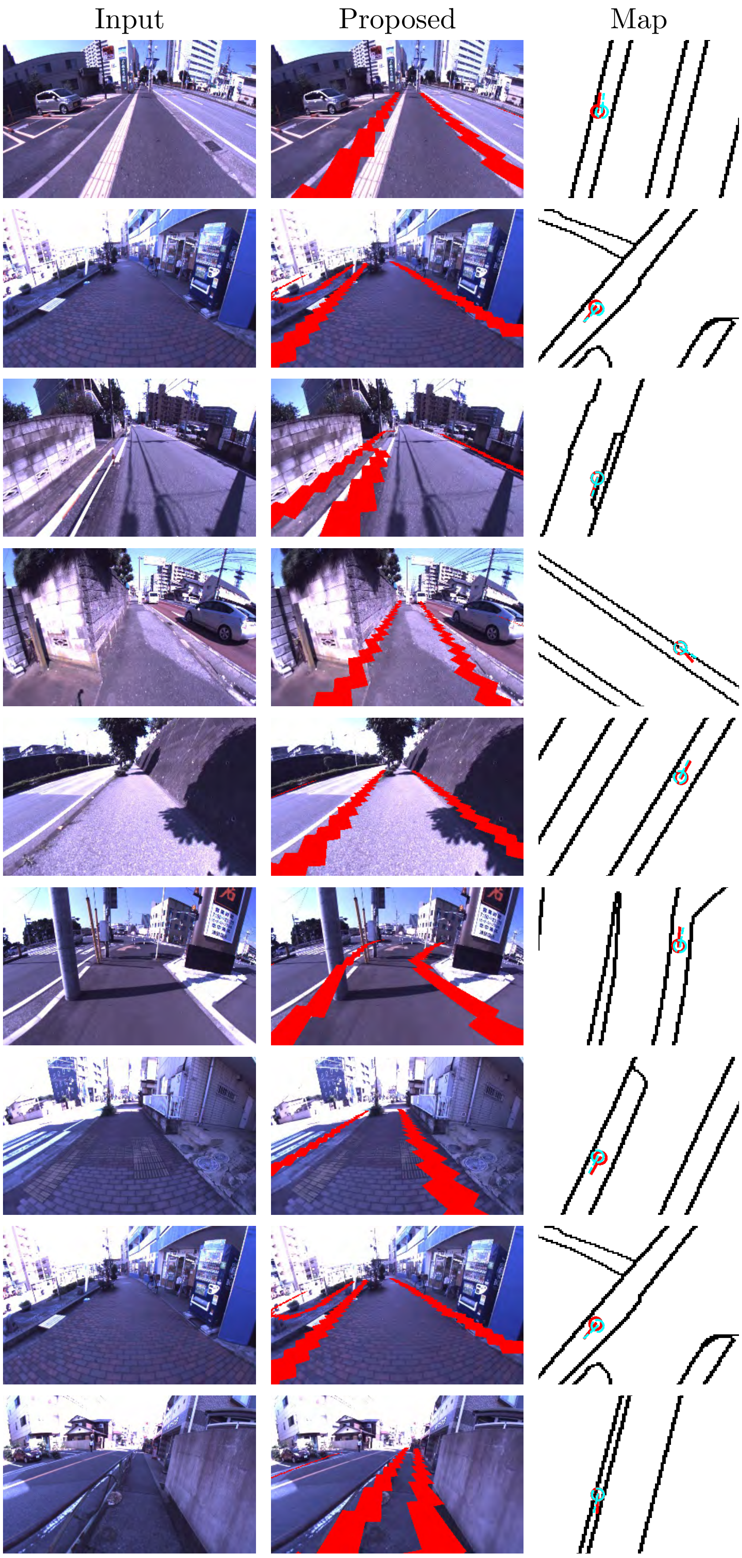


Figure 3.8: Sample successful results of single image localization experiments using dataset B. In the maps the estimated pose and ground-truth positions are indicated in red and blue, respectively.

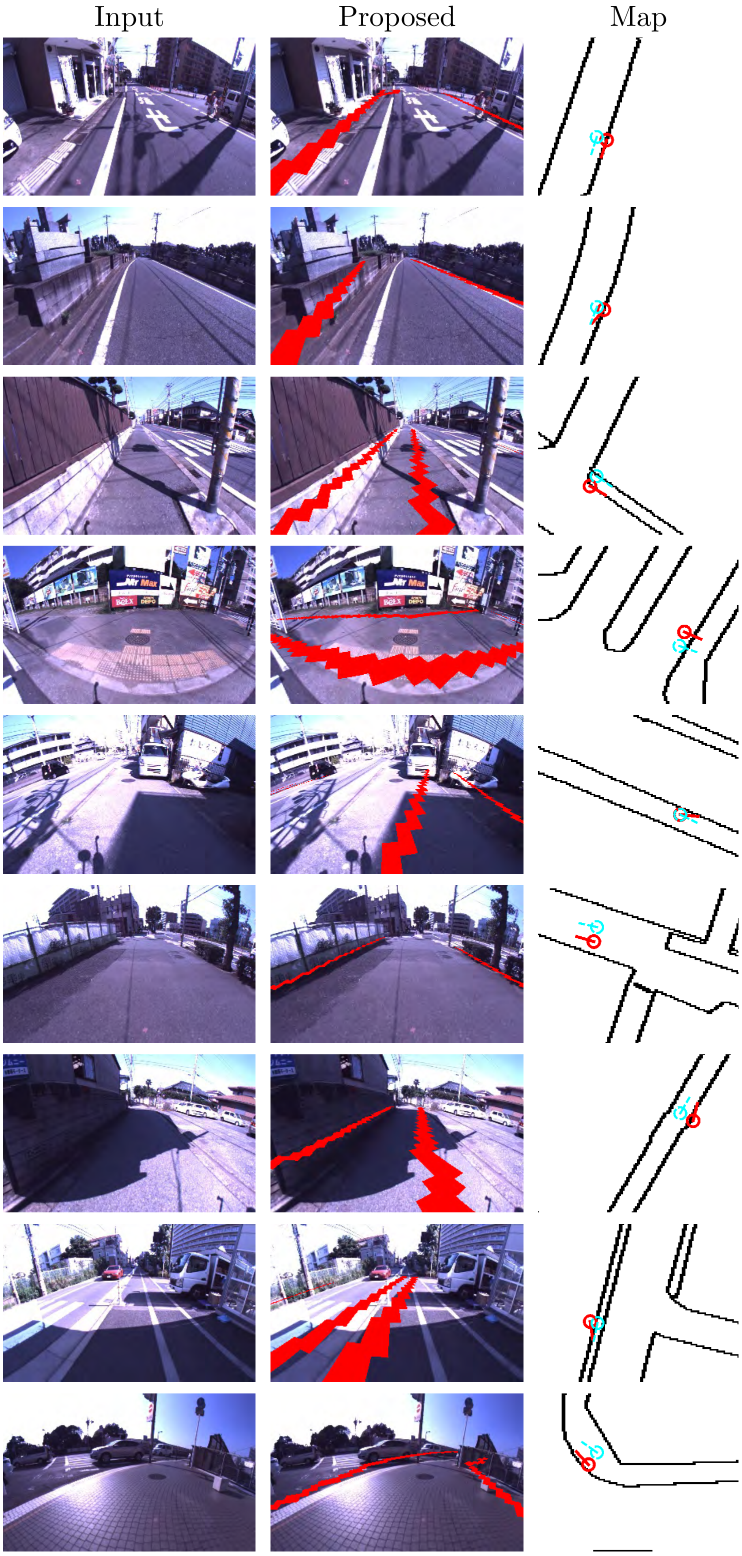


FIGURE 3.9: Sample unsuccessful results of single image localization experiments using dataset B.

Table 3.3: Comparisons of the four different dependence measures. Mean position and orientation errors (± SD) for dataset A are shown. Run time is measured per pose hypothesis on a laptop computer with a Core-i7 5500U.

Measure	Position error [m]	Orientation error [°]	Run time [s]	Estimator
SMI	0.45 ±0.42	4.3 ±4.5	0.26 ±0.007	LSMI [Sakai and Sugiyama, 2014]
MI	1.04 ±0.67	11.9 ±7.0	6.7 ±0.56	MLMI [Suzuki *et al.*, 2009b]
QMI	0.77 ±0.70	7.0 ±7.0	0.30 ±0.03	LSQMI [Sainui and Sugiyama, 2013]
SCE	1.75 ±0.63	19.5 ±5.6	1.4 ±0.28	LSCE [Tangkaratt *et al.*, 2015]
dCor	1.05 ±0.79	5.9 ±5.8	0.038 ±0.012	Empirical dCor [Székely *et al.*, 2007]

Comparison of Image Size In the experiments using dataset A, images of 320×240 pixels were used to extract features. They were then downsampled to 80×60 pixels. We compared the performance of our system for different downsampling ratios and the results are summarized in Figure 3.12. It is apparent that the localization accuracy was not largely affected even when downsampled by a factor of 5 (both in the height and width directions, resulting in 64 × 48 pixels).

3.4.3 Position Tracking

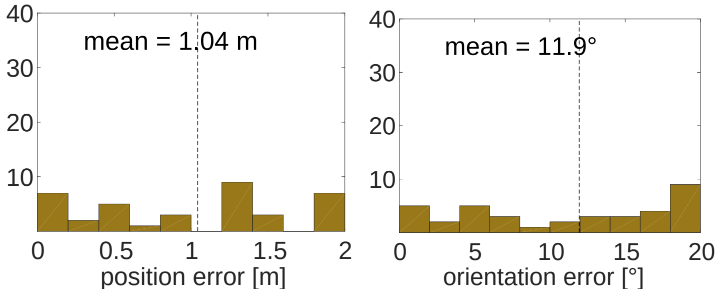
We integrated our method with a particle filter and conducted position tracking experiments.

3.4.3.1 Setup

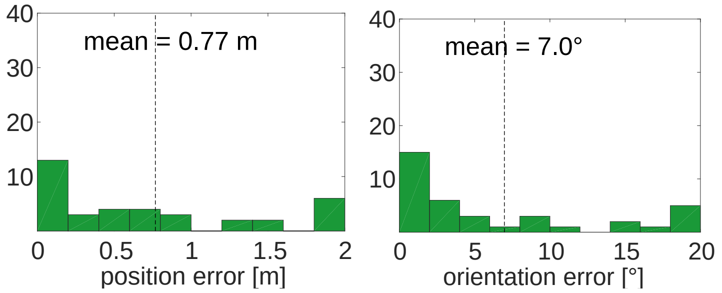
Position tracking experiments were executed using datasets collected by manually navigating a wheel-chair robot (Figure 3.13). The datasets include LiDAR (Velodyne HDL-32e) measurements, images from an omnidirectional camera (Pointgrey Ladybug3), gyro-assisted odometry, and real time kinematic (RTK) GPS data for reference. Localization experiments were conducted offline using the collected data.

Gyro-assisted odometry was used for the particle-filter prediction. The update step was executed every 2 s (0.5-Hz frequency) using the local-grid-map measurement model depicted in Figure 3.14. In each update step, a local grid map was generated using the point cloud and camera images accumulated since the previous update step. The six-dimensional features listed in Table 3.4 were extracted for each cell in the grid map. They are denoted by $\{\boldsymbol{x}_i\}_{i=1}^{n}$ (n denotes the number of cells in the local grid map). After extracting the features, the particles were weighted using SMI and resampled. The input data pairs $\{(\boldsymbol{x}_i, y_i^{(\boldsymbol{w})})\}_{i=1}^{n}$ were obtained by matching the local grid map with the street map using the robot pose hypothesis $\boldsymbol{w}$.

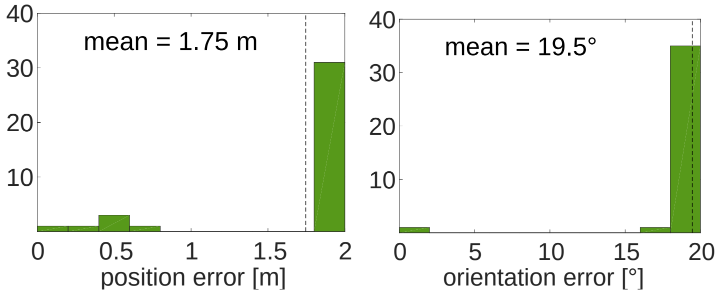
The details of the feature extraction procedure are as follows. Each point in the cloud was assigned a color (RGB) and gradient magnitude (the response of the Sobel filter) feature by projecting the point onto the image plane. As geometric features, we employed the elevation of the point and laser scan discontinuity as described in [Levinson and Thrun, 2013]. The laser scan



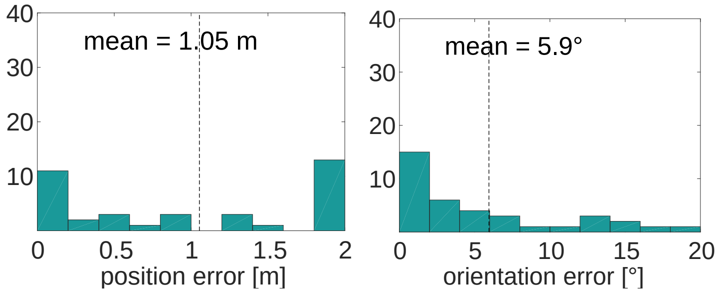
MI



QMI

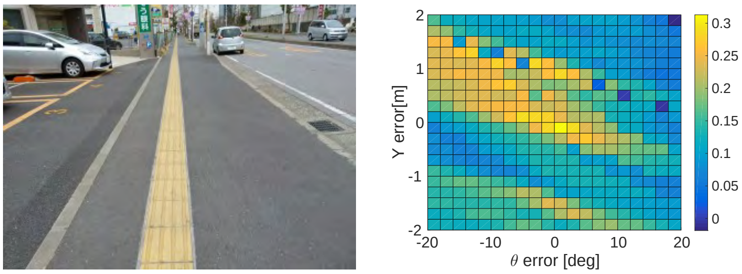


SCE

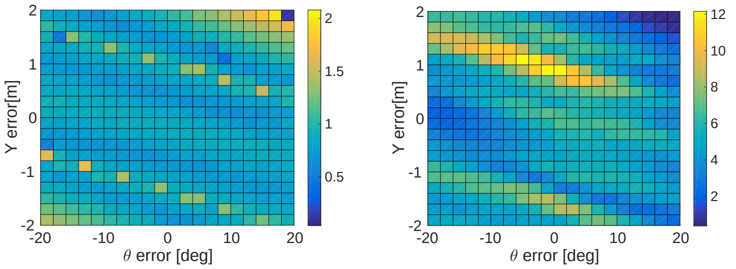


dCor

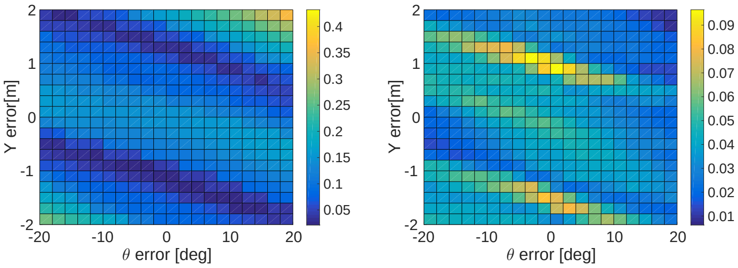
FIGURE 3.10: Histograms for comparisons of dependence measures (results with dataset A).



Input SMI



MI QMI



SCE dCor

FIGURE 3.11: Comparison of score distributions for different dependence measures.

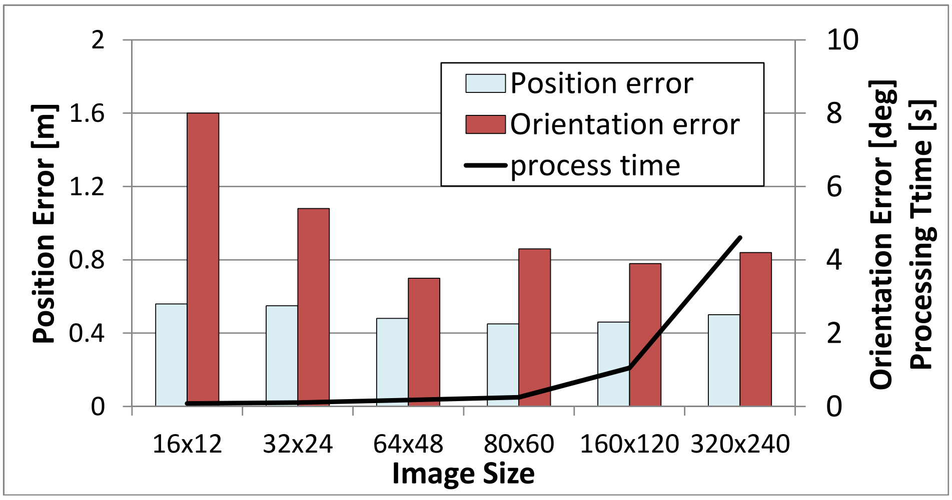


Figure 3.12: Performance comparison for different image downsampling ratios. The mean position and orientation errors and the processing time per position hypothesis are shown.

FIGURE 3.13: Robot used to collect datasets for position tracking experiments. It is equipped with a 3D LiDAR, an omnidirectional camera, a monocular camera, a gyro and a GPS receiver.

discontinuity was calculated for each point p_i in the point cloud, using

$$\max(R_{p_{i-1}} - R_{p_i}, R_{p_{i+1}} - R_{p_i}, 0)^{0.5}, \tag{3.6}$$

where the R_{p_i} denotes the LiDAR range measurement for the point p_i, and p_{i-1} and p_{i+1} are the neighboring points. After the above-mentioned point-wise feature extraction, the points were projected onto the ground plane and stored in the grid cells. The cell-wise features $\boldsymbol{x}$ were obtained by averaging the point-wise features and scaling them to the range between 0 to 1.

Some other details of the evaluation setup are as follows. The size of the local grid maps were 10 m×10 m and the cell size was 10 cm. To reduce the processing time, the grid map was downsampled by a factor of 4 before SMI estimation. The number of particles was 100, and the number of kernel centers used to estimate the SMI was 40. Unless otherwise noted, the initial particle distribution was given by a normal distribution with a standard deviation of 1 m and 5°, and the cross-validation omission rate was set to 90%. A laptop computer with a Core i7-4710HQ CPU was used to measure the processing time. Our implementation was written in MATLAB and was not optimized for multi-core CPUs.

Table 3.4: Features used in position tracking experiments.

Category	Feature	Dimension
Image color	color (R, G, B)	3
Image gradient	Sobel magnitude	1
Height	elevation of the points	1
Height change	laser scan discontinuity	1

3.4.3.2 Comparative Method

Because no existing method can be directly compared to ours, we implemented a comparative method based on road boundary detection and matching. We incorporated the idea of road edge detection using height changes and image edges [Matsushita and Miura, 2011]. The comparative method employed the same particle filter used in the proposed method, except for the likelihood calculation. The likelihood for each particle was approximated by the correlation between the road boundaries in the street map and the road boundary likeliness of the local grid cells as shown below:

$$p(\{\boldsymbol{x}_i\}_{i=1}^n|\boldsymbol{w},\boldsymbol{M}) = \frac{\sum_{i=1}^n f_{\mathrm{local}}(\boldsymbol{x}_i) f_{\mathrm{map}}(y_i^{(\boldsymbol{w})})}{\sqrt{\sum_{i=1}^n f_{\mathrm{local}}(\boldsymbol{x}_i)^2}\sqrt{\sum_{i=1}^n f_{\mathrm{map}}(y_i^{(\boldsymbol{w})})^2}}, \tag{3.7}$$

$$f_{\mathrm{local}}(\boldsymbol{x}) = x^{(\mathrm{g})} \cdot x^{(\mathrm{d})}, \tag{3.8}$$

$$f_{\mathrm{map}}(y) = \begin{cases} 1 & \text{if } y \text{ is road boundary,} \\ 0 & \text{otherwise.} \end{cases} \tag{3.9}$$

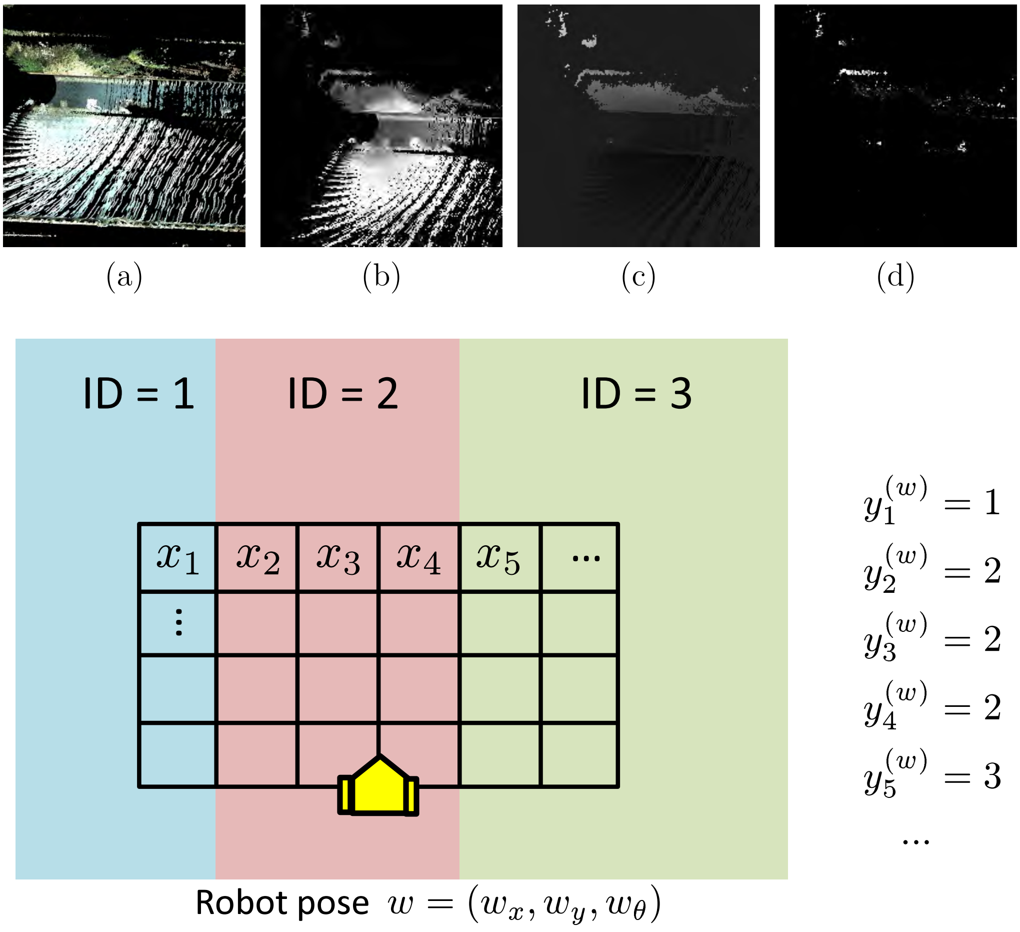
Here, $x^{(\mathrm{g})}$ and $x^{(\mathrm{d})}$ denote the image gradient feature and laser discontinuity feature, respectively.

3.4.3.3 Results

Here, we show results obtained using two datasets C and D, which were collected in Narashino, Chiba, Japan. The duration and distance of the navigation was 127 s and 150 m for dataset C, and 3,514 s and 4.5 km for dataset D, respectively.

Localization Accuracy We conducted position tracking experiments 100 times for each dataset using different random seeds. Both exemplary results and camera images are shown in Figures 3.15 and 3.16. In both figures, it can be seen that our method successfully corrected the odometry errors and tracked the robot position, while the comparative method resulted in large errors for dataset D.

The localization accuracy was evaluated based on the root mean squared (RMS) error using GPS as the ground-truth. Table 3.5 and Figure 3.17 summarize the results. The mean RMS error of the proposed method over 100 runs was 0.6 m (approximately 2.6 pixels on the map) for dataset C and 1.33 m for dataset D. The errors were significantly smaller ($p < 0.01$ according to the paired t-test) than those of the comparative method. Note that the estimation errors include both GPS and map errors. Figure 3.18 shows an example of map errors observed at (D) in Figure 3.16. The difference between the road position in the satellite image and on the map was approximately 2 m. Nevertheless, our method "correctly" tracked the robot position along the road indicated on the map.



(a) (b) (c) (d)

(e)

FIGURE 3.14: Sample of local grid map generation and measurement model. (a) 3D point cloud obtained by a LiDAR and cameras. A grid map is generated from the point cloud and features stored in the grid map are shown in (b-d). (b) Color feature (red channel). (c) Height feature. (d) Laser discontinuity feature. (e) Illustrative example of data association between a grid map and the map.

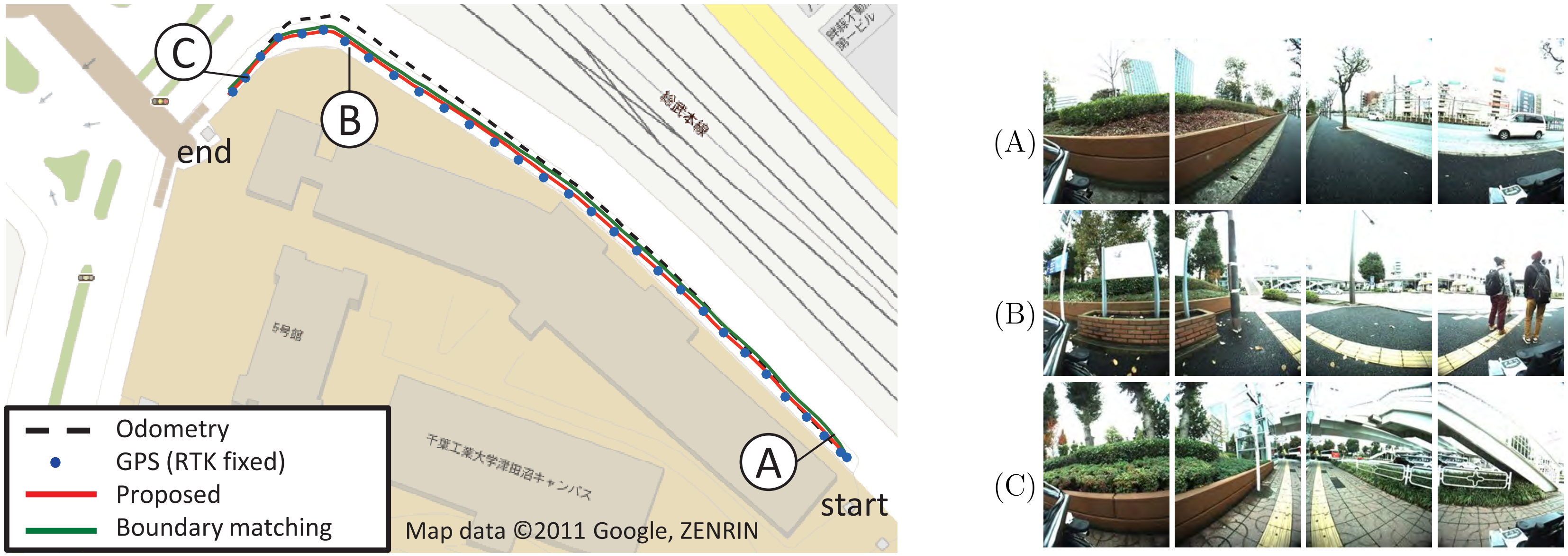


FIGURE 3.15: Position tracking result for dataset C (150 m, 127 s).

Table 3.5: Mean RMS error (± SD in meters) over 100 runs of the position tracking experiments.

	proposed method	boundary matching
dataset C	0.60 ± 0.13	1.2 ± 0.78
dataset D	1.33 ± 0.10	9.58 ± 20.7

Comparison of Cross-Validation Omission We compared the computational time of the particle-filter update and the localization accuracy for different omission rates. For each omission rate, we conducted position tracking 100 times using dataset D. Figure 3.19 (a) summarizes the results. By omitting up to 90% of the cross-validation procedures, the average processing time decreased from 1.8 s to 0.3 s, without significant degradation of the localization accuracy.

Robustness Against Odometry Errors The robustness against odometry errors was evaluated using distorted odometry logs. We generated distorted odometry logs by adding Gaussian noise with different noise levels. For each noise level, 100 distorted odometry logs using different random seeds were generated. We compared the ratio of successful localizations (where the final position error was within 2 m). The results are summarized in Figure 3.19 (b). As can be seen from that figure, our method showed superior robustness compared to the comparative method. Further, our system maintained good localization accuracy even with relatively large odometry errors. Accuracy degradation began when the standard deviation of the noise exceeded 20% and 2 °/s for the distance and orientation, respectively.

Robustness Against Initial Position Errors The robustness against initial position errors was evaluated using inaccurate initial positions. We randomly shifted the mean of the initial particle distribution by 1–6 m and conducted position tracking experiments. The standard deviations of the initial particle distribution were matched with the position error sizes. We evaluated the ratio of successful localizations using 100 differently shifted initial positions for each error size. The results are summarized in Figure 3.19 (c). The proposed method maintained a 100% success rate for initial position errors of up to 1 m. Although the localization accuracy degraded as the error level

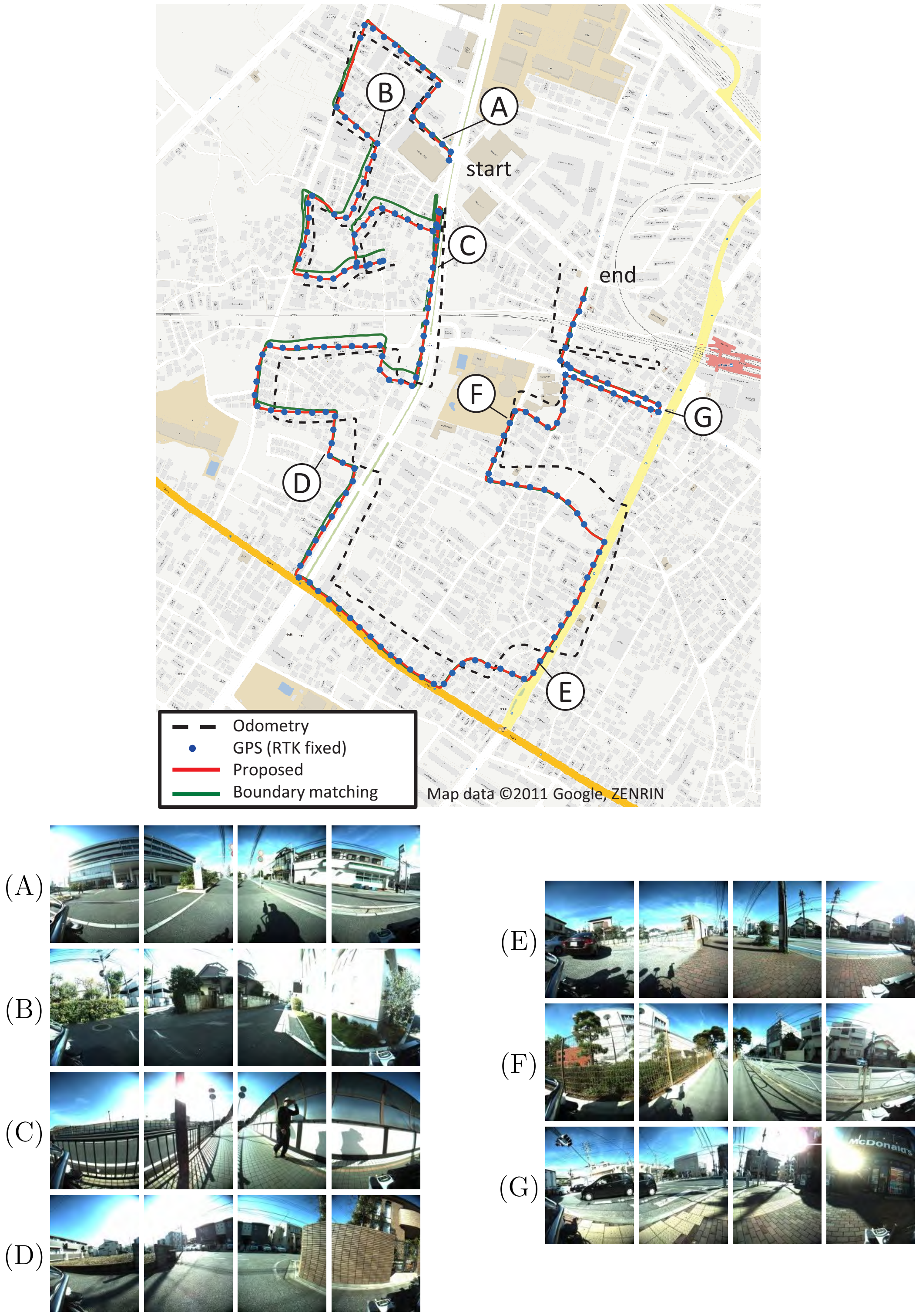
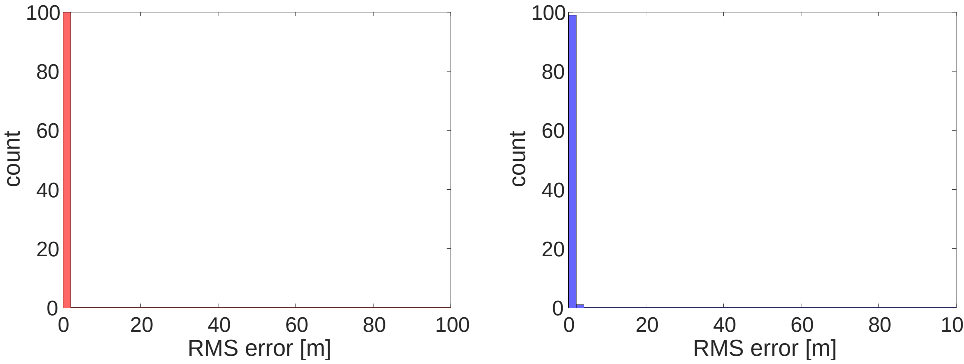
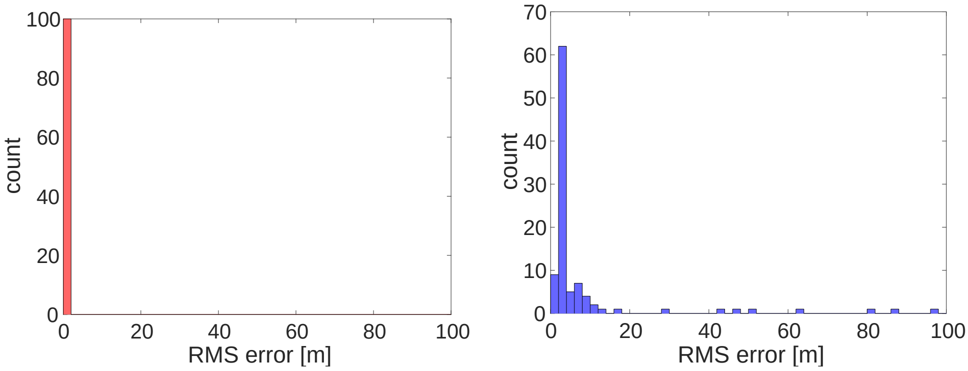


FIGURE 3.16: Position tracking result for dataset D (4.5 km, 3,514 s). The proposed method took 725 sec (excluding feature extraction time) to estimate the trajectory.

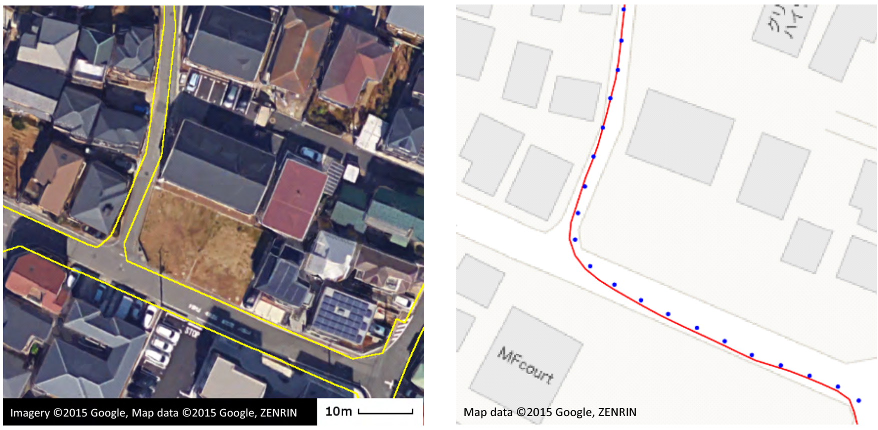


Dataset C - proposed method   Dataset C - boundary matching



Dataset D - proposed method   Dataset D - boundary matching

Figure 3.17: Error histograms of position tracking experiments.



(a) (b)

Figure 3.18: Sample map errors. (a) Satellite image with road boundary line overlay from (b) street map. A notable difference in road shape can be seen at the intersection.

Table 3.6: Comparison of localization accuracy between LSMI and BLSMI. Mean RMS errors (± SD) in meters are shown.

	BLSMI	LSMI (proposed)
dataset C	0.61 ± 0.14	0.60 ± 0.13
dataset D	1.33 ± 0.12	1.33 ± 0.12

increased, the proposed method was superior to the comparative method in terms of the degradation speed.

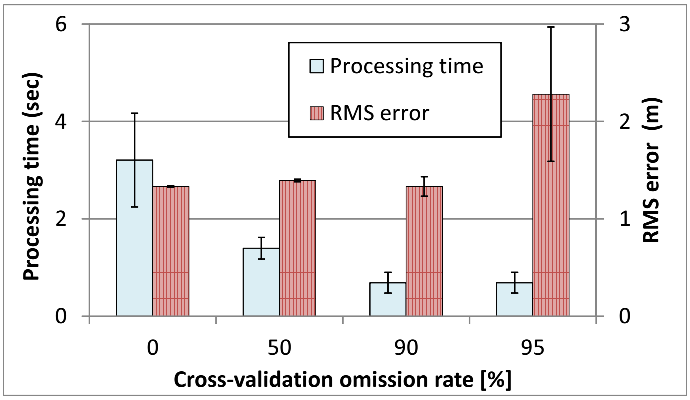
3.4.4 Efficacy of BLSMI

We conducted additional experiments to verify whether BLSMI is effective for localization.

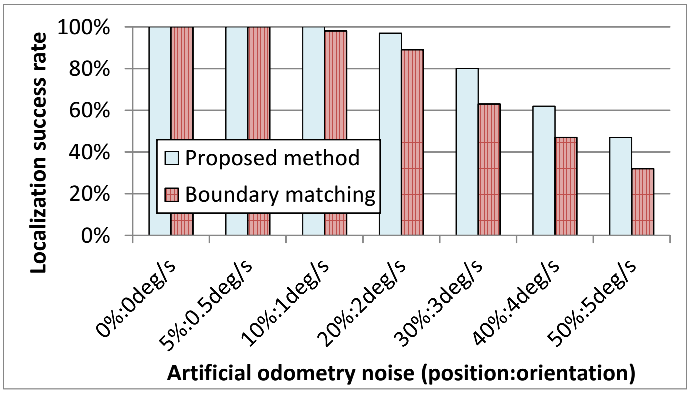
Single Image 2-degrees-of-freedom Localization We conducted the same experiments as shown in Section 3.4.2. The results using dataset A are shown in Figure 3.20. The differences between LSMI and BLSMI were non-significant ($p = 0.80$ for position errors and $p = 0.74$ for orientation errors by the paired t-test).

Position Tracking We also conducted position tracking experiments using BLSMI. The setup of the experiments was identical to Section 3.4.3. RMS errors over 100 runs are summarized in Table 3.6. The differences between LSMI and BLSMI were not significant. As can be seen from Figure 3.21, the processing time of BLSMI is significantly larger than LSMI.

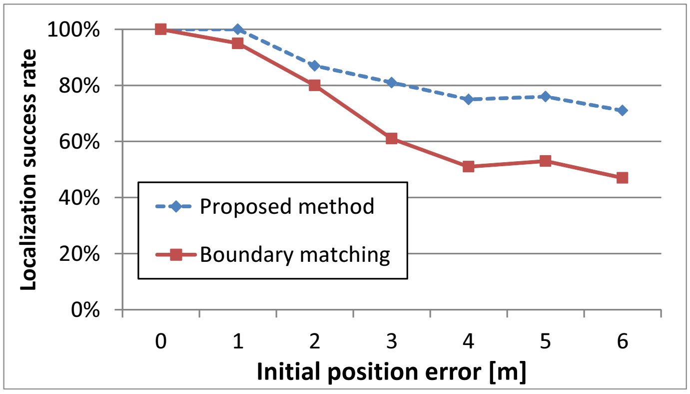
Discussion As can be seen from the above results, employing BLSMI did not improve the accuracy of localization. A possible reason for that is that gradient calculation is not involved in our method. While BLSMI was developed aiming to resolve local-optima problems in maximization of LSMI, our method employs a particle filter and no gradient-based optimization is used; local optima can be avoided through random resampling. On the other hand, BLSMI has disadvantages in terms of processing time. Therefore, we conclude that the ordinary LSMI is more suitable than BLSMI for this localization problem.



(a)

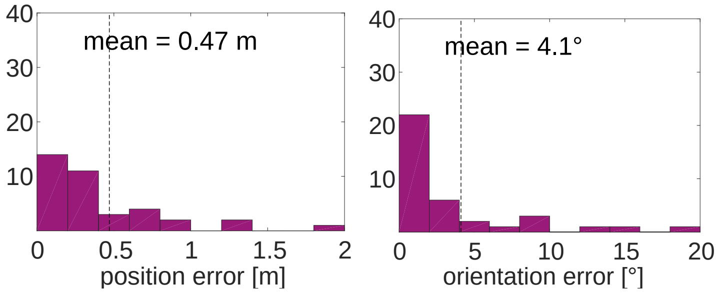


(b)

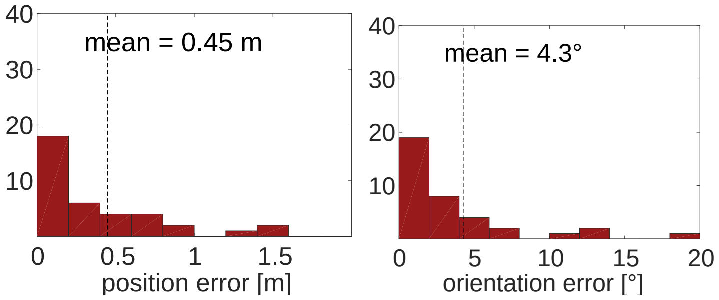


(c)

Figure 3.19: (a) Comparison of update time and localization errors for different cross-validation omission rates (the time for generating local grid maps is excluded). The error bars indicate standard deviations. (b) Comparison of localization success rate using distorted odometry logs. (c) Comparison of localization success rate with initial position errors.



BLSMI



LSMI (proposed)

FIGURE 3.20: Single image localization experiments using BLSMI.

3.5 Conclusions

This chapter described a robot localization method based on a 2D street map. The key issue we tackled was determining the correct correspondence between the 2D street map and sensor measurements. The proposed method finds the correspondence to localize a robot through maximization of the statistical dependence between segments of the map and features extracted from the sensor data. This dependence is measured using the squared-loss variant of a mutual information estimator, which is robust and computationally efficient.

The performance of our method was evaluated through extensive experiments using images and LiDAR data collected in real-world urban environments. The proposed method exhibited reasonable localization accuracy. We also observed that SMI exhibited the best performance among several examined dependence measures in terms of both localization accuracy and processing time. Note that the robustness against occlusions and map error management of this method should be improved through further research.

As dependence maximization localization functions without road boundary

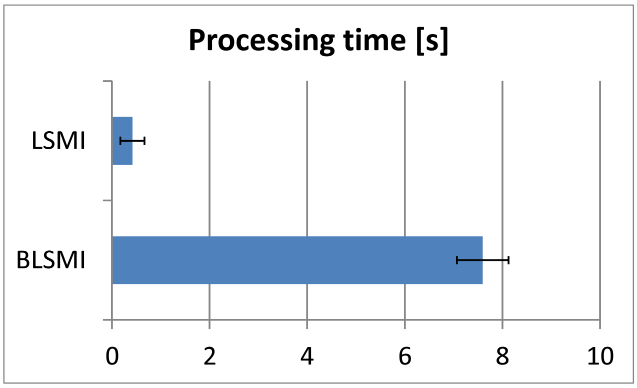


FIGURE 3.21: Comparison of update time. Error bars indicate standard deviations.

detection, this approach should be particularly useful for environments that are not as structured as urban roadways. The successful localization results obtained without using any prior knowledge of the environment allow us to hope that our method can be easily applied to *new* environments. Finally, because our method is general and not restricted to the road network, it should be applicable to localization of other types of moving objects such as flying robots, indoor mobile robots and pedestrians.

Chapter 4

Target-less Multi-modal Sensor Calibration

4.1 Introduction

Combining a 3D LiDAR and cameras is a popular means of acquiring RGB combined with depth (RGB-D) information. Such a sensor combination is particularly useful for outdoor mobile robots. While Kinect-type RGB-D cameras that directly outputs colored 3D points are typically employed for indoors, they are not capable for outdoors. The reason is that their range of measurements is small (typically ∼5 m) for the size of the environment, and they are ineffective under direct sunlight [Zennaro *et al.*, 2015]. To assign image colors to LiDAR points correctly, identifying the geometric relationship between the sensors is necessary. This study considers the problem of extrinsic calibration (i.e., estimation of the rigid-body transform) between a 3D LiDAR and monocular camera using sensor data.

Extrinsic calibration is a general concern in using multiple sensors. Calibration problems are often formulated as registering multiple sensor data. Although two sensor data of the same modality (e.g., two color images) are relatively easy to register, registering multi-modal sensor data (e.g., a point cloud and color image) is harder because we must identify correspondences among completely different information.

4.1.1 Target-less Calibration

To overcome the difficulty of correspondence, calibration targets designed to enable easy finding of correspondences have been used [Zhang and Pless, 2004]

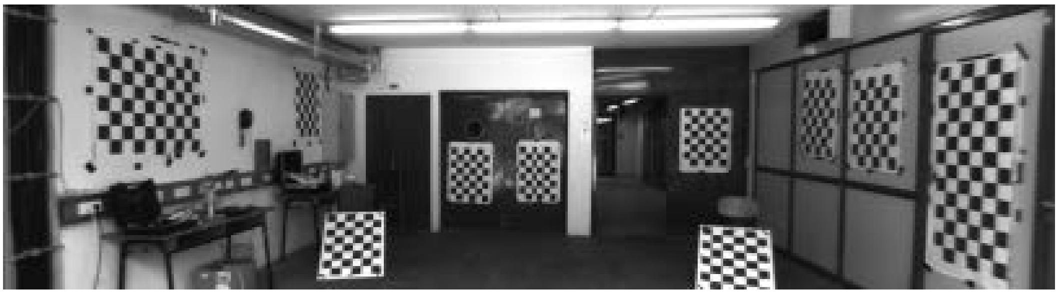


FIGURE 4.1: Sample input for calibration using checkerboard targets [Geiger *et al.*, 2012].

[Kwak *et al.*, 2011] [Alismail *et al.*, 2012] [Geiger *et al.*, 2012]. A typical choice for calibration target is a board with checker pattern (Figure 4.1). Recently, several calibration methods that do not require special targets (so called *target-less* calibration) have been proposed [Pandey *et al.*, 2012] [Taylor and Nieto, 2012] [Napier *et al.*, 2013]. While calibration of a system is typically performed during the production process, recalibration is also important for long-life systems because sensor configurations can change over time. Target-less calibration can reduce the cost of preparing calibration targets and enable on-site and on-line recalibration.

4.1.2 Calibration Approaches

Existing target-less calibration methods can be classified into two categories by their registration approach: edge-based and dependence-based. The method proposed by Levinson and Thrun [Levinson and Thrun, 2013] is a seminal work on edge-based calibration. Discontinuities in LiDAR scans and image edges are matched together to evaluate the calibration quality. The underlying assumption is that changes in the distance or depth are likely to appear as edges in an image [Baboud *et al.*, 2011]. While Levinson and Thrun employed only the strength of the edges [Levinson and Thrun, 2013], Taylor *et al.* reported the usefulness of the orientation of edges [Taylor *et al.*, 2013]. They proposed using gradient orientation measure that can evaluate the degree to which edge orientations are aligned between a camera image and LiDAR reflectivity image.

In dependence-based methods, calibration is performed by maximizing a dependence metric. The assumption is that multiple sensor data that is observed from the same object should have dependence. For example, reflectivity of LiDAR measurements (the intensity of laser return) tends to be high on white objects and vegetation, and low on dark-colored objects. This implies that considerable dependence between reflectivity and image color exists. Several registration methods using the property have been proposed [Williams *et al.*, 2004] [Mastin *et al.*, 2009]; Pandey *et al.* as well as Taylor and Nieto applied the idea to calibration [Pandey *et al.*, 2012] [Taylor and Nieto, 2012]. Although the method of Pandey *et al.* maximizes MI using gradient ascent, we observed that it often got stuck in local optima with outdoor data sets as a result of the non-smoothness of the objective function.

4.1.3 Contribution

In this chapter, a novel dependence-based calibration method is proposed. The major component we employ is bagged least-squares mutual information (BLSMI) that allows us to incorporate more features than previous methods. The increased number of features helps to construct a considerably smooth objective function. We tested our method by a series of comparative experiments and observed significant improvements in accuracy and robustness.

4.2 Proposed Method

Our proposed method can be regarded as an extension of existing dependence-based calibration methods [Pandey *et al.*, 2012] [Taylor and Nieto, 2012]. Table 4.1 summarizes the differences between our own and previous methods. Here, we describe the proposed method and provides detailed comparisons.

Table 4.1: Comparison of dependence-based calibration methods. The dimensionalities of features are given in parentheses.

	Metric	Optimization	LiDAR feature	Image feature
[Pandey *et al.*, 2012]	MI	steepest gradient descent	reflectivity (1)	grayscale intensity (1)
[Taylor and Nieto, 2012]	NMI	particle swarm optimization	surface normal (1)	grayscale intensity (1)
Proposed method	SMI	quasi-Newton (BFGS)	reflectivity (1) discontinuity (1) surface normal (3)	color (3) edge strength (1)

4.2.1 Overview of Dependence-based Calibration

A point cloud collected by a LiDAR and an image captured by a camera are used as input. The number of total points in the cloud is denoted by n and features extracted on the points are denoted by real vectors $\{\boldsymbol{x}_i\}_{i=1}^n$. The correspondence between the LiDAR data and camera image is made by projecting the 3D points onto the image plane. By using the 6-DoF relative camera pose with respect to the LiDAR coordinate frame $\boldsymbol{\Theta} = [t_x, t_y, t_z, r_x, r_y, r_z]$, we calculate the projected pixel location (u_i, v_i) for each 3D point.

Image features extracted at (u_i, v_i) in the image are denoted by real vectors $\{\boldsymbol{y}_i\}_{i=1}^n$. The LiDAR and image features are denoted by random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. Under the assumption that data pairs $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ are generated from the joint probability $p(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta})$, we maximize an objective function f that measures statistical dependence between $\boldsymbol{X}$ and $\boldsymbol{Y}$, to determine the calibration parameter $\boldsymbol{\Theta}$.

$$\widehat{\boldsymbol{\Theta}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} f(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta}). \tag{4.1}$$

In the previous methods, several statistical dependence measures including MI and normalized MI (NMI) have been employed as f. Because f is generally non-convex and the optimization is not trivial, various methods have been applied to solve this maximization problem [Pandey *et al.*, 2012] [Taylor and Nieto, 2012].

4.2.2 Features

While different sets of features have been employed in the existing dependence-based calibration methods [Pandey *et al.*, 2012] [Taylor and Nieto, 2012], we have found that using a richer set of features is important for improving the robustness against varying environments. Therefore, our method uses a combined set of features (Table 4.1).

Each of the features employed previously has both advantages and disadvantages. The image grayscale intensity and laser reflectivity (intensity of laser return) employed in Pandey *et al.* [Pandey *et al.*, 2012] are less useful in outdoor scenes. For outdoor calibration, Taylor and Nieto [Taylor and Nieto, 2012] reported the effectiveness of surface normal information extracted from

LiDAR measurements. We have found that the combination of these complementary features improves calibration accuracy on both indoor and outdoor datasets.

In addition, we have found that incorporating the edge-based features used in the method proposed by Levinson and Thrun [Levinson and Thrun, 2013] improve the calibration accuracy. The underlying assumption of their method is that depth changes are likely to appear as edges in an image [Baboud *et al.*, 2011]. The assumption can also be regarded as the existence of dependence between the discontinuity and edge features.

We present detailed feature extraction procedures in Section 4.3.1.

4.2.3 Optimization

We employ the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method [Kelley, 1987], which is a gradient-based algorithm, to maximize (4.1). Because the dependence function is not always smooth, a gradient search algorithm can easily get stuck in a local maximum. To overcome this problem, we follow the data aggregation approach presented in [Pandey *et al.*, 2012]. By aggregating many laser scans and images collected in different places, we can obtain a smooth objective function.

A different approach to overcoming local maxima was taken in [Taylor and Nieto, 2012]. They employed the particle swarm optimization (PSO) algorithm, which is a heuristic search used to find the global optimum of a non-convex problem. However, the computational cost of PSO is high because a large number of samples are employed during the search. Therefore, they accelerated their implementation by GPU.

4.2.4 Dependence Measure

We employ SMI as a dependence measure. Although previous methods estimate discretized versions of MI and NMI using joint histogram-based estimators for computational efficiency [Pandey *et al.*, 2012] [Taylor and Nieto, 2012], as discussed in Chapter 2, such histogram-based estimators are strongly affected by the growth of dimensionality and cannot be used with high-dimensional data. For example, if the total dimensionality of the features is four, a four-dimensional histogram must be constructed. If we discretize each feature into 256 steps and count the frequency using 32-bit integers, $256^4 \times 4\text{bytes} = 16\text{GB}$ of memory is required. Because our method uses nine-dimensional features, implementing the estimator on current computer systems is impossible (this kind of phenomenon is known as *the curse of dimensionality* [Bellman, 1961]).

To alleviate the issue of the dimensionality, we estimate the dependence directly in the continuous space using LSMI [Sakai and Sugiyama, 2014], which is a kernel-based, computationally efficient SMI estimator.

4.2.5 Bagged LSMI

To further improve the smoothness of the objective function, we employ the bagged LSMI (BLSMI) presented in Chapter 2. By using BLSMI, we can obtain much more smooth objective functions compared to using the original LSMI; therefore, we can avoid the local optima problem in a gradient-based search.

4.3 Experimental Setup

We employed Multisense SL (Figure 4.2) from Carnegie Robotics for evaluation because its factory calibrated parameters are available. Multisense SL consists of a spinning 2D LiDAR (UTM-30LX-EW by Hokuyo Automatic) and stereo camera. In the following experiments, we estimated the relative pose between the LiDAR and left camera (right camera images were not used).

4.3.1 Details of Feature Extraction

4.3.1.1 UTM-30LX-EW Reflectivity Calibration

Hokuyo UTM-30LX-EW can output a reflectivity response for each distance measurement. The reflectivity value depends on the observed objects, especially on their material and surface orientation, and also on the distances to the objects. While the material and surface orientation should have strong

Figure 4.2: Multisense SL used in experiments. The 2D LiDAR mounted on a spindle rotates continuously.

dependence on image color features, the distance to the object has little dependence. Therefore, the dependence between reflectivity and distance can hinder identifying the dependence between measurements from a LiDAR and camera.

To alleviate the effect of the distance, we normalized reflectivity regarding the distance. We collected reflectivity data of a white paper measured from a perpendicular direction with different distances. Collected measurements are shown in Figure 4.3. We employed the least-squares method and determined that the following function fits well to the data:

$$\begin{aligned} g(x) &= a\exp(bx) + c\exp(dx), \\ \text{where } a &= 4207,\ b = -0.06993, \\ c &= 2.814 \times 10^4,\ d = -1.981. \end{aligned}$$

The normalized reflectivity is calculated using the raw reflectivity measurements $\{r_i\}_{i=1}^n$ and the corresponding distance measurements $\{d_i\}_{i=1}^n$ as follows.

$$r_i^{\text{normalized}} = \frac{r_i}{g(d_i)}.$$

Exemplary raw and normalized reflectivity measurements are presented in Figure 4.4 (a) and (b).

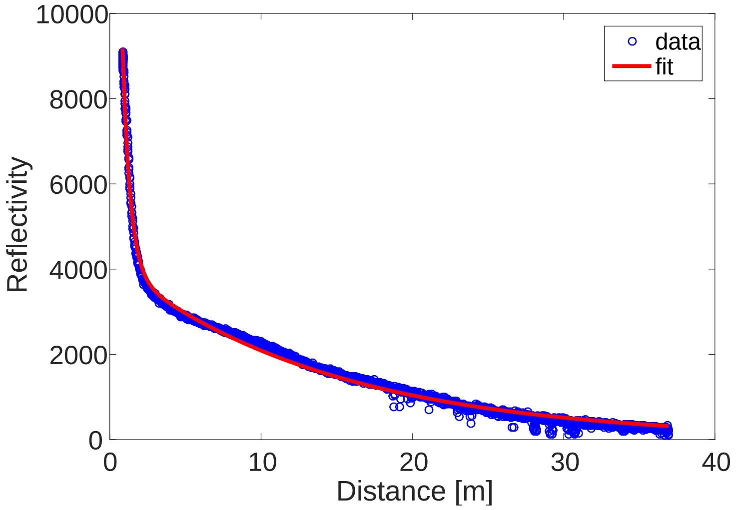


Figure 4.3: Reflectivity response of Hokuyo UTM-30LX-EW against a white paper (blue dots) and a fitted curve (red line).

4.3.1.2 LiDAR Scan Discontinuity

The LiDAR discontinuity feature is calculated as described in [Levinson and Thrun, 2013]. Let $d_1, ..., d_L$ be the distance measurements in a single (2D) scan. The discontinuity feature e^{LiDAR} for the i-th measurement is calculated by:

$$e_i^{\text{LiDAR}} = \max(d_{i-1} - d_i, d_{i+1} - d_i, 0)^{\gamma}.$$

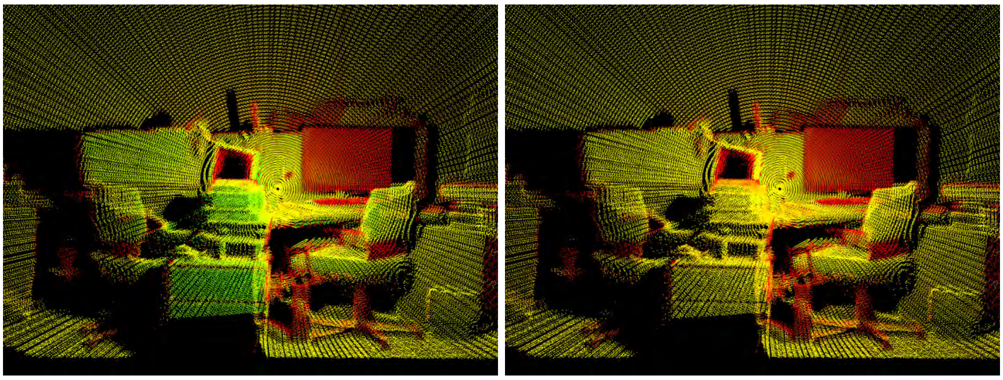
We used $\gamma = 0.5$ in our experiments. The max operator is used because closer points at the discontinuity are more likely to coincide with image edges. An example of an extracted discontinuity feature is given in Figure 4.4 (c).

4.3.1.3 Surface Normal

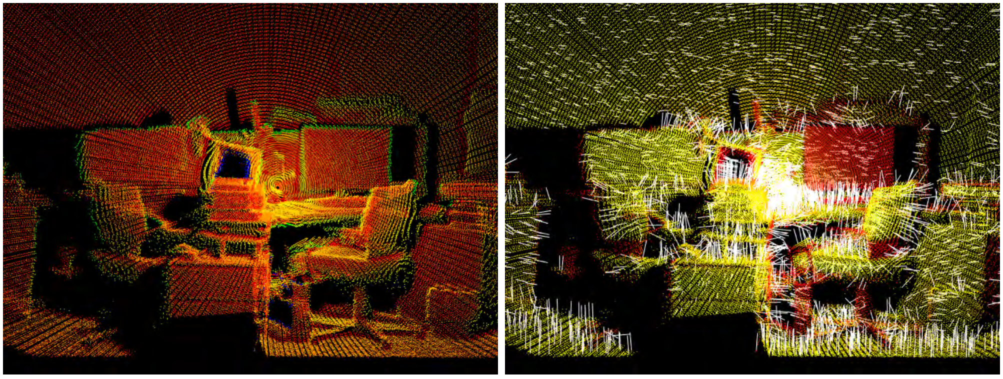
Surface normals were extracted as follows. For each 3D point, we extracted 100 nearest-neighbor points and applied least-squares plane fitting to the following plane equation.

$$ax + by + cz + d = 0.$$

The estimated coefficients a, b, c were used as three-dimensional features. We used the implementation included in the Point Cloud Library [Rusu and Cousins, 2011] to calculate the plane parameters. Figure 4.4 (d) shows an example of surface normal detection.



(a) Raw reflectivity (b) Normalized reflectivity



(c) LiDAR discontinuity (d) Surface normals

FIGURE 4.4: Sample features extracted from LiDAR measurements. Green indicates higher and red indicates lower reflectivity/discontinuity. White lines in (d) indicates surface normal vectors.

4.3.1.4 Image Features

In our experiments, the image size of Multisense SL's camera was 1024×544. RGB pixel values were used as three-dimensional color features. Another image feature employed was edge strength, which was calculated as described in [Levinson and Thrun, 2013]. For each pixel, grayscale intensity was compared with eight-neighbor pixels and the maximum intensity difference (non-negative) was used as a one-dimensional feature:

$$e^{\text{image}}_{i,j} = \max_{x,y\in\{-1,0,1\}} |I_{i,j} - I_{i+x,j+y}|,$$

where $I_{i,j}$ denotes the grayscale intensity for pixel (i, j). To ensure the smoothness of the objective function, we applied a Gaussian filter to the extracted image features. A Gaussian width $\sigma = 2$ was used in our experiments.

The extracted features were stored pixel-wise. However, when we project 3D points onto an image, the projected points do not exactly correspond to single pixels (i.e., coordinates are not always integer). We have found that sub-pixel interpolation considerably improves the smoothness of the objective function. Instead of simply rounding the coordinates into integer, we calculated sub-pixel image features using the following linear interpolation. The feature for image location (u, v) is calculated by:

$$\begin{aligned} &(1-\beta)\left((1-\alpha)\boldsymbol{f}_{\lfloor u\rfloor,\lfloor v\rfloor} + \alpha\boldsymbol{f}_{\lceil u\rceil,\lfloor v\rfloor}\right) \\ &\quad + \beta\left((1-\alpha)\boldsymbol{f}_{\lfloor u\rfloor,\lceil v\rceil} + \alpha\boldsymbol{f}_{\lceil u\rceil,\lceil v\rceil}\right), \\ &\text{where} \quad \alpha = u - \lfloor u\rfloor, \quad \beta = v - \lfloor v\rfloor, \end{aligned}$$

and, $\boldsymbol{f}_{i,j}$ denotes the image feature for pixel (i, j).

4.3.2 Comparative Methods

We implemented the following comparative methods.

Pandey *et al.* **[Pandey *et al.*, 2012]**

Grayscale intensity of the image and reflectivity of LiDAR measurements were used as input features. MI was employed as the objective function f and estimated in the following steps. First, $p(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{\Theta})$ was estimated by KDE in a discretized space (256×256). Our implementation of the KDE was constructing a 2D joint histogram and applying a Gaussian filter to it [1]. Discretized MI was then calculated as shown in (2.3).

Taylor and Nieto [Taylor and Nieto, 2012]

Taylor and Nieto proposed using surface normals calculated from LiDAR measurements as a feature [Taylor and Nieto, 2012]. The one-dimensional surface normal feature was calculated for each 3D measurement in a single 2D scan $\{(p_i^x, p_i^y, p_i^z)\}_{i=1}^{L}$ using the following equation:

$$\eta_i = \frac{(p_i^y - p_{i+1}^y)^2}{\sqrt{(p_i^x - p_{i+1}^x)^2 + (p_i^z - p_{i+1}^z)^2}}.$$

---

[1] We followed the implementation publicly available in [Pandey, 2015].

Grayscale intensity was used as the image feature. NMI, which is calculated as follows, was employed as the objective function:

$$\mathrm{NMI}(\boldsymbol{X},\boldsymbol{Y}) = \frac{\mathrm{H}(\boldsymbol{X})+\mathrm{H}(\boldsymbol{Y})}{\mathrm{H}(\boldsymbol{X},\boldsymbol{Y})}. \tag{4.2}$$

The joint probability was estimated similarly to that in [Pandey *et al.*, 2012].

Discretized SMI

This method was introduced to demonstrate the difference of dependence measures. In this method, the same features and method described in [Pandey *et al.*, 2012] were employed to estimate the histogram of the joint probability. The following discretized SMI was calculated from the histogram.

$$\mathrm{SMI}(\boldsymbol{X},\boldsymbol{Y}) = -\frac{1}{2}\sum_{x}\sum_{y} p(\boldsymbol{x})p(\boldsymbol{y})\left(\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}-1\right)^2. \tag{4.3}$$

BLSMI (proposed method)

In our proposed method, we employed all the features shown in Table 4.1. Bootstrap data sets for BLSMI were generated by randomly subsampling 1% of the total input points (for example, when 20 scans were aggregated, 2,000 points were extracted for each bootstrap set). To reduce the computation time, cross-validation in LSMI estimation was performed only on the first bootstrap data set and hyper-parameters were reused for all other data sets. In our preliminary experiments, omitting cross-validation drastically reduces the processing time without degrading calibration accuracy.Unless otherwise noted, the number of bagging replications (M in (2.9)) was set to 200.

LSMI without bagging

This method is exactly the same as the proposed method except that bagging was not performed. Single LSMI estimation using all input data was used as the objective function f.

Levinson and Thrun [Levinson and Thrun, 2013]

While our experimental setup is batch calibration in which single calibration is performed using the aggregated data, the method by Levinson and Thrun checks calibration accuracy and update the parameters on-the-fly [Levinson and Thrun, 2013]. Therefore, we slightly modified their objective function for application to the batch calibration problem. The following batch version of their objective function was employed:

$$J_c = \frac{1}{n} \sum_{i=1}^{n} e_i^{\mathrm{LiDAR}} \cdot e_i^{\mathrm{image}}. \tag{4.4}$$

Here, e_i^{image} is the image edge strength that corresponds to the i-th laser measurement. Maximizing the objective function (4.4) can be interpreted as maximizing *linear* dependence between LiDAR discontinuity and image edge strength. By contrast, our method employs an MI-based measure to handle *non-linear* dependence.

4.4 Results

We collected data in indoor and outdoor scenes and evaluated our method in two ways: qualitative evaluation of the objective functions and quantitative evaluation of the calibration accuracy. The factory calibrated parameters of the Multisense SL were used as the ground truth.

4.4.1 Datasets

We collected 3D scans of the LiDAR and camera images from 19 indoor and 20 outdoor scenes. Because the importance of nearby objects was pointed out by Pandey *et al.* [Pandey *et al.*, 2012], we included some outdoor scenes with nearby objects (Figure 4.5). Each scene contained a single 3D scan (360° rotation of the 2D LiDAR) and a color image.

Although Multisense SL's 3D LiDAR has a considerably larger field of view (FoV) than its camera, points out of the camera's FoV are not useful for calibration. To reduce the computational cost, we reduced the number of points by limiting the FoV of the 3D LiDAR to 60° × 40°. In addition, we randomly extracted 10,000 points from each scene. The points extracted from different scenes were combined together and used as the input for calibration.

4.4.2 Comparison of Objective Functions

As preliminary experiments, we investigated the objective functions of the proposed and previous methods. Exemplary objective functions around the ground truth are shown in Figure 4.6 and 4.7. The objective function employed in Pandey *et al.* [Pandey *et al.*, 2012] appeared smooth for indoor data sets, but it exhibited highly non-smooth curvature for outdoor data sets. A possible cause of the difference is noises in the data. In outdoor environments, objects were not completely motionless (e.g., tree branches were swaying in the wind). In addition, over and under exposures caused by strong sunlight would interfere in the dependence between laser reflectivity and image intensity. By contrast, the objective function of the proposed method appeared to have smooth curvatures and exhibited single maxima for both data sets.

4.4.3 Evaluation of Calibration Accuracy

We evaluated the calibration accuracy of our method compared to others. Note that we used only features and dependence measures from the previous methods. The same BFGS search was applied to all methods for comparison. We implemented BFGS search using the `fminunc` function in MATLAB [The MathWorks, Inc., 2015] and the gradients were calculated numerically. To evaluate the stability, we conducted calibration experiments 100 times using different input data. At the beginning of each experiment, points were randomly extracted (again, 10,000 points per scene). To simulate errors by a hand-measured calibration, the parameters were initialized by adding uniform random noise (±3 cm and ±3°) to the ground truth parameters. The results are summarized in Figure **??**, showing that the proposed method exhibited the smallest repeatability error.

The quantitative calibration accuracy was measured using projection errors. 3D points in the data sets were projected onto the image plane using the estimated calibration parameters. We calculated the 2D position difference between points projected using estimated parameters and of that using the ground truth parameters. Figure 4.9 summarizes the results. The mean projection error of our method was 3.1 pixels using 19 indoor scans and 4.6 pixels using 20 outdoor scans, and they were significantly smaller ($p < 0.01$ by the paired t-test) than any other comparative methods. On outdoor data sets, the method of Pandey *et al.* [Pandey *et al.*, 2012] often got stuck in local optima and resulted in degraded performance.

Exemplary colored point clouds obtained using the calibration results (with 20 outdoor scans) are shown in Figure 4.10. The quality of the colored point clouds appeared to be competitive to the ground truth. Therefore, we consider the estimated parameters to be sufficient to acquire colored 3D points.

4.4.4 Comparison of Features

As we integrated several features, the contribution of each feature is of our interest. We compared calibration accuracy using the different subset of LiDAR features. We again executed calibration experiments 100 times for each subset of features, using BLSMI as the objective function. Figure 4.11 summarizes the result. It can be seen that single feature is not enough for indoor-outdoor calibration; reflectivity and discontinuity were not effective on outdoor data sets and surface normal performed poorly on indoor data sets. By combining these features, we were able to compensate the weakness and also improve the calibration accuracy.

We found that reasonable calibration accuracy can be obtained without using reflectivity. Therefore we consider our method can be used even with LiDARs that do not return reflectivity.

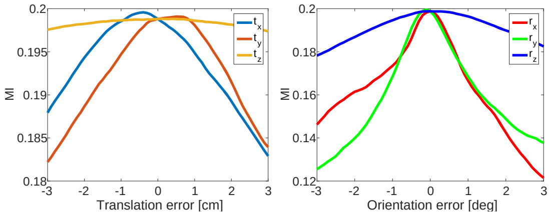
4.4.5 Comparison of the Number of Bagging Replications

We evaluated the performance of our method for the different number of bagging replications (M in (2.9)). The results are summarized in Figure 4.12. Although $M = 200$ was used in the experiments in the previous sections, $M = 50$ and $M = 100$ also seemed to provide reasonable accuracy with much less processing time.

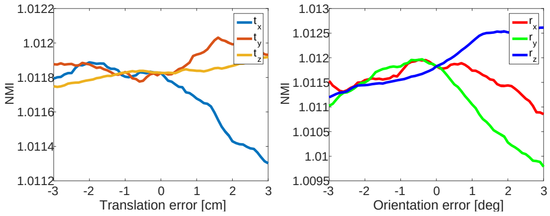
indoor data

outdoor data

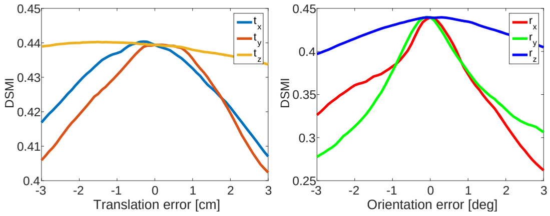
Figure 4.5: Sample input data used in experiments. Top: indoor scenes. Bottom: outdoor scenes. Some of outdoor scenes contained nearby objects.



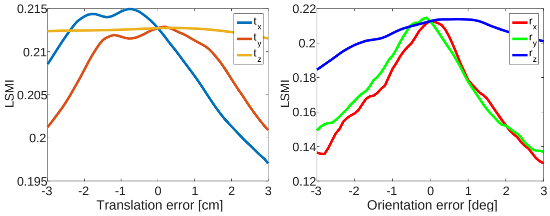
Pandey *et al.* (2.3)



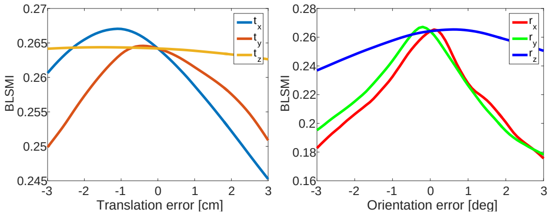
Taylor and Nieto (4.2)



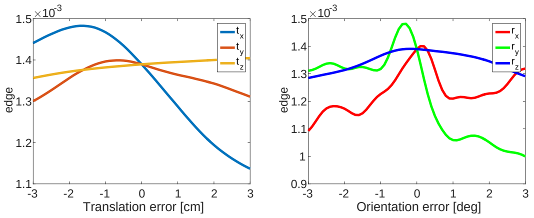
Discretized SMI (4.3)



LSMI without bagging (2.8)

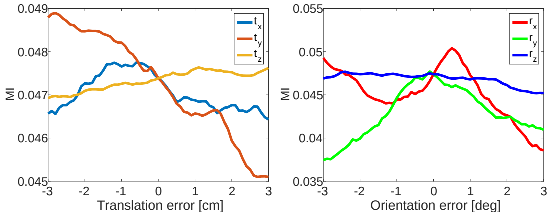


BLSMI (2.9) (*proposed*)

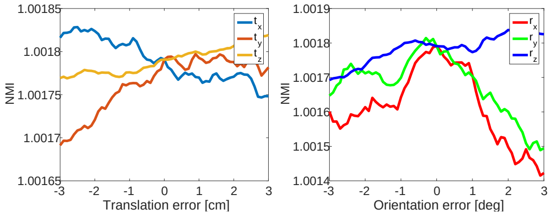


Levinson and Thrun (4.4)

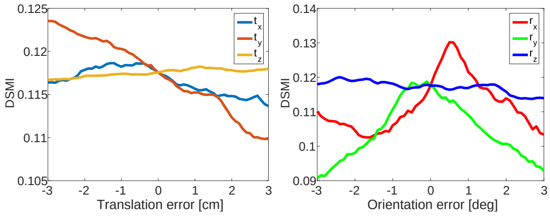
FIGURE 4.6: Sample visualization of objective functions for indoor datasets.



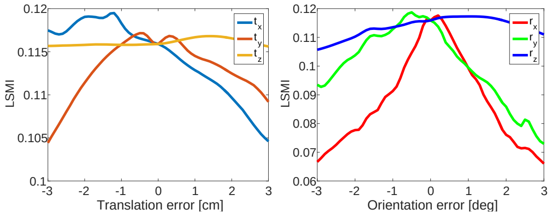
Pandey *et al.* (2.3)



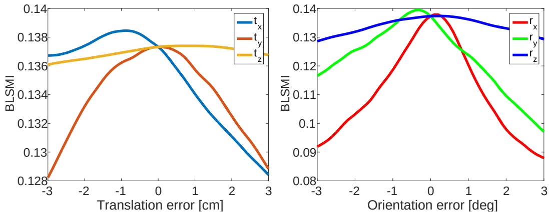
Taylor and Nieto (4.2)



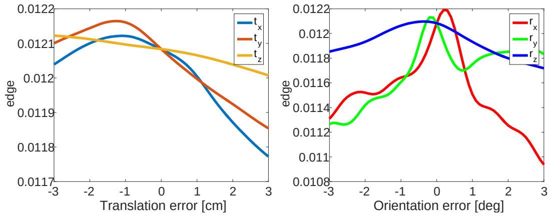
Discretized SMI (4.3)



LSMI without bagging (2.8)

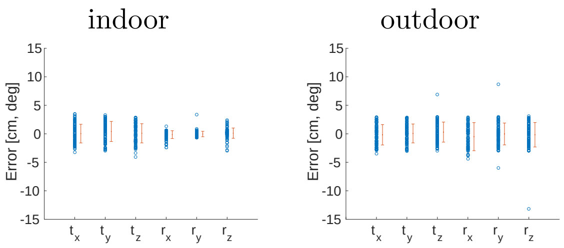


BLSMI (2.9) (*proposed*)

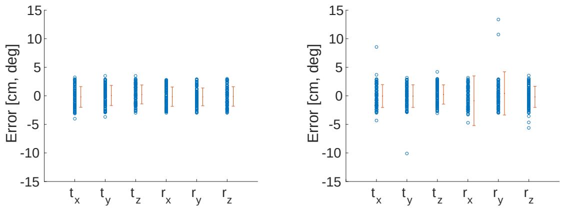


Levinson and Thrun (4.4)

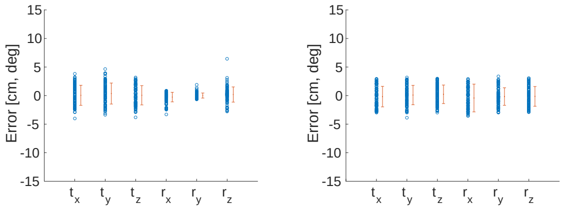
FIGURE 4.7: Sample visualization of objective functions for outdoor datasets.



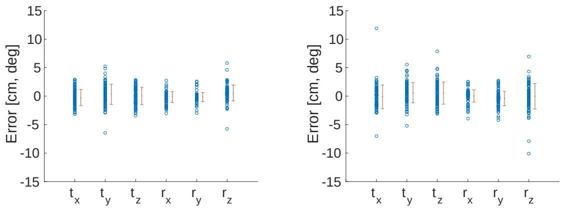
Pandey *et al.* (2.3)



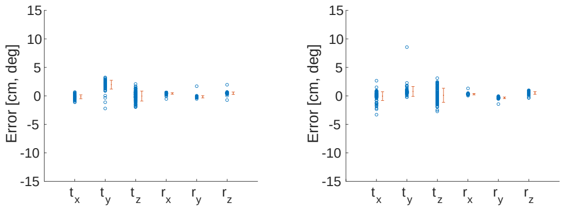
Taylor and Nieto (4.2)



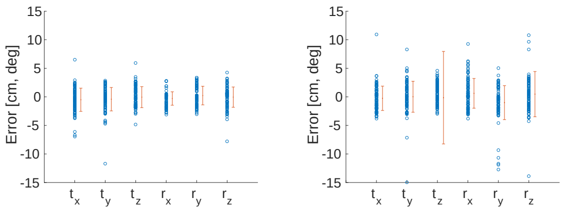
Discretized SMI (4.3)



LSMI without bagging (2.8)



BLSMI (2.9) (*proposed*)



Levinson and Thrun (4.4)

Figure 4.8: Summary of calibration experiments over 100 runs. Error bars show standard deviations.

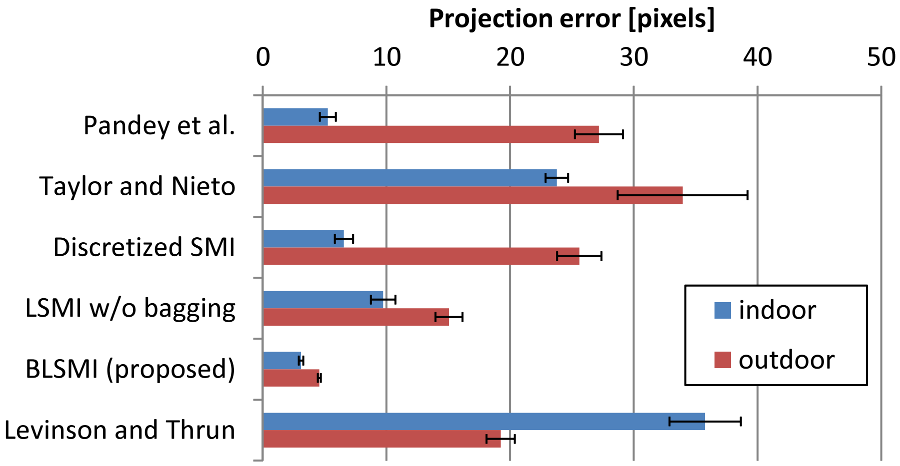


FIGURE 4.9: Comparison of projection error using different features and dependence estimators. Error bars indicate standard errors. Our proposed method exhibited the smallest projection error. Pandey *et al.* [Pandey *et al.*, 2012] showed degraded performance on outdoor data sets because it got stuck in local optima.

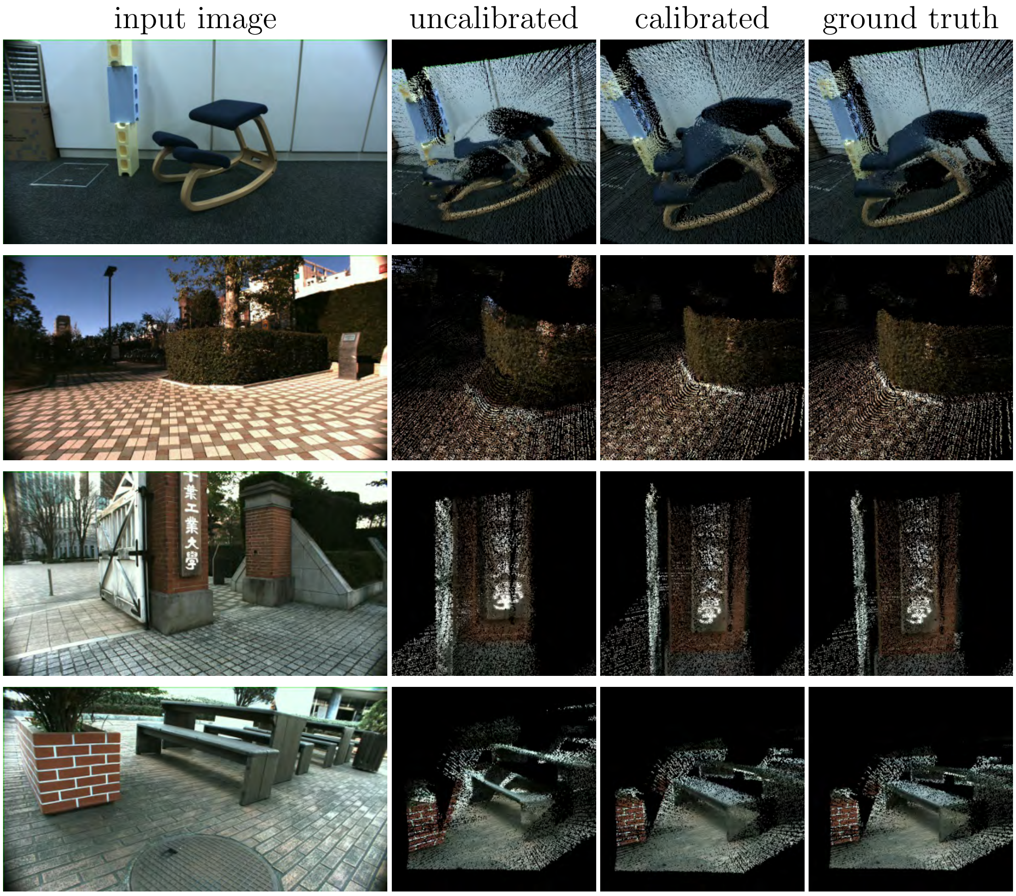


FIGURE 4.10: Sample calibration results.

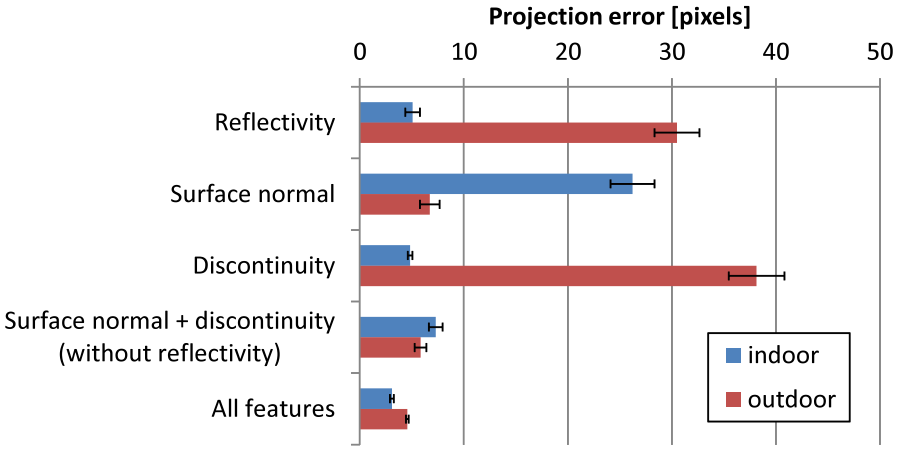


FIGURE 4.11: Comparison of calibration accuracy using different subset of LiDAR features. BLSMI was used as the objective function. Error bars indicate standard errors.

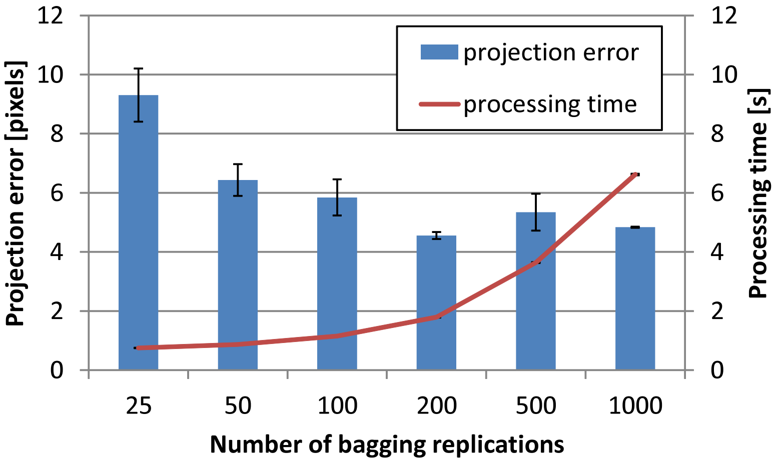


Figure 4.12: Comparison of projection error and processing time per dependence evaluation using different number of bagging replications (20 outdoor scans were used). Reasonable calibration accuracy was obtained by 50 or more replications. Error bars indicate standard errors.

4.5 Conclusion

In this chapter, a novel extension of dependence-based calibration methods was proposed. The major issue we tackled was the non-smoothness of the objective function, which caused the previous methods to get stuck in the local optima. Richer features had to be incorporated to address the problem. However, the previously employed dependence estimators were not feasible to handle the increased dimensional features. Therefore, we introduced a novel dependence estimator, BLSMI. Consequently, we obtained a smooth objective function, and an efficient gradient-based search was successfully applied.

We compared our proposed method with previous methods by experiments using indoor and outdoor data sets. While our method showed the best calibration accuracy on all the data sets we employed for evaluation, the improvement was particularly significant on outdoor data sets. Therefore, our method should be useful for robots navigating outdoors.

Chapter 5

Conclusion and Future Work

In this chapter, conclusions and a discussion on further research directions are presented.

5.1 Conclusions

This thesis was devoted to extending a robot's ability of navigation aiming to realize navigation in unrehearsed environments. Two issues closely related to navigation were considered.

- Localization based on existing street maps.
- Target-less multi-modal sensor extrinsic calibration.

Both of them were addressed using the common framework of maximizing statistical dependence.

In Chapter 3, the issue of mobile robot localization using a 2D street map was considered. It was found out that localization can be achieved by maximization of squared-loss mutual information (SMI) between sensor observations and the map. Based on this finding, we developed a novel localization method. This enables localization in sidewalks, for which no effective localization method has been known before. Reducing the computation cost for estimating SMI that was another challenge for practical use, we developed a computationally efficient implementation that achieved near real-time position tracking performance using a laptop-class computer. Experiments using real-world datasets were conducted to validate the effectiveness of the method. Successful localization results were obtained including 4.5 km position tracking experiments.

In Chapter 4, the issue of automatic, target-less extrinsic calibration between a LiDAR and a camera was considered. This issue was addressed by finding the relative pose that maximizes statistical dependence between LiDAR data and camera images. Although there already exists a method that utilizes MI for this purpose, it is rather noise sensitive and the objective function to maximize tends to be non-smooth on noisy outdoor data; therefore, it is easily got trapped in local optima. To solve the problem, the use of richer features is promising; however, the previously used dependence estimator was not feasible to handle high dimensional features. To cope with the problem, we proposed combining an SMI estimator with bootstrap aggregating. It was experimentally shown that the objective function using the bootstrapped SMI estimator was smooth even on noisy outdoor data, and the developed calibration method outperformed several existing methods in accuracy.

As described above, we presented solutions to two important problems regarding navigation. While our contributions are only parts of the long journey toward navigation in unrehearsed environments, we made a significant step forward. We hope someday in the future, when our goal is achieved by continued research, navigation technology will be widely used all over the world.

5.2 Future Work

Here, several proposals for future research are discussed.

5.2.1 Improvements on Feature Extraction

In both calibration and localization, we employed rather simple features. We could improve the performance by applying feature learning methods [Bengio *et al.*, 2013]. While handling image features should be straightforward because deep convolutional neural networks have been successfully applied to image recognition problems [A. Krizhevsky and Hinton, 2012] [Girshick and Malik, 2014], further investigation should be required for LiDAR features because feature learning of depth or 3D shape information has not been not fully established.

5.2.2 Dealing with Significant Occlusions

As discussed in Chapter 2, we can see the robustness of SMI from its mathematical definition. Also, we actually observed its high performance in the calibration and localization experiments. Nevertheless, we encountered several failures in localization experiments caused by occlusions or outliers. It should be useful in practice if we can detect such failures or evaluate the localization confidence from data. Machine learning methods on anomaly detection [Chandola *et al.*, 2009] [Hido *et al.*, 2011] could be applied. To handle occlusions caused by moving objects, techniques employed in conventional SLAM [Wangsiripitak and Murray, 2009] [Tomono, 2009] should be applicable to our method.

5.2.3 Real World Navigation Experiments

Lastly but not least, performing navigation experiments using a real robot equipped with the developed methods is an necessary step. Through off-line experiments, we observed that our localization method achieved good localization accuracy with short processing time. Nevertheless, on-line experiments will definitely help us to find out more challenges that we might not be aware of yet.

Bibliography

[A. Krizhevsky and Hinton, 2012] A. Krizhevsky, I. S. and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the Neural Information Processing Systems Conference (NIPS)*, Montreal, Canada.

[Alempijevic *et al.*, 2009] Alempijevic, A., Kodagoda, S., and Dissanayake, G. (2009). Cross-modal localization through mutual information. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 5597–5602, St. Louis, Missouri, USA.

[Alismail *et al.*, 2012] Alismail, H. S., Baker, L. D., and Browning , B. (2012). Automatic calibration of a range sensor and camera system. In *2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission (3DIMPVT)*.

[Baboud *et al.*, 2011] Baboud, L., Cadik, M., Eisemann, E., and Seidel, H.-P. (2011). Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *Proceedings of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, pages 41–48.

[Bellman, 1961] Bellman, R. E. (1961). *Adaptive control processes - A guided tour*. Princeton University Press.

[Bengio *et al.*, 2013] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

[Bertozzi *et al.*, 2000] Bertozzi, M., Broggi, A., and Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32(1):1–16.

[Brakatsoulas *et al.*, 2005] Brakatsoulas, S., Pfoser, D., Salas, R., and Wenk, C. (2005). On map-matching vehicle tracking data. In *Proceedings of the*

31st International Conference on Very Large Data Bases (VLDB), pages 853–864, Trondheim, Norway.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[Brubaker *et al.*, 2013] Brubaker, M. A., Geiger, A., and Urtasun, R. (2013). Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, pages 3057–3064, Portland, Oregon, USA.

[Buehler *et al.*, 2009] Buehler, M., Iagnemma, K., and Singh, S., editors (2009). *The DARPA Urban Challenge*. Springer.

[Chandola *et al.*, 2009] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58.

[Chausse *et al.*, 2005] Chausse, F., Laneurit, J., and Chapuis, R. (2005). Vehicle localization on a digital map using particles filtering. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 243–248, Las Vegas, Nevada, USA.

[Collignon *et al.*, 1995] Collignon, A., Maes, F., Delaere, D., Vandermeulen, D., Suetens, P., and Marchal, G. (1995). Automated multimodality image registration using information theory. In *Proceedings of 14th International Conference on Information Processing in Medical Imaging (IPMI)*, pages 263–274, Ile de Berder, France.

[DARPA, 2007] DARPA (2007). DARPA Grand Challenge '05. `http://archive.darpa.mil/grandchallenge05/`.

[Dyson, 2016] Dyson (2016). Robot cleaner dyson 360 eye. 2016. [Accessed: 20- May- 2016].

[Floros *et al.*, 2013] Floros, G., van der Zander, B., and Leibe, B. (2013). OpenStreetSLAM: Global vehicle localization using OpenStreetMaps. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1054–1059, Karlsruhe, Germany.

[Geiger *et al.*, 2012] Geiger, A., Moosmann, F., Car, O., and Schuster, B. (2012). Automatic camera and range sensor calibration using a single shot. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 3936–3943.

[Girshick and Malik, 2014] Girshick, R. B., D. J. D. T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, pages 580–587.

[Guizzo, 2011] Guizzo, E. (2011). `http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works`. [Accessed: 20- May- 2016].

[Haklay and Weber, 2008] Haklay, M. M. and Weber, P. (2008). OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18.

[Hatao *et al.*, 2009] Hatao, N., Hanai, R., Yamazaki, K., and Inaba, M. (2009). Real-time navigation for a personal mobility in an environment with pedestrians. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 619–626.

[Hentschel and Wagner, 2010] Hentschel, M. and Wagner, B. (2010). Autonomous robot navigation based on OpenStreetMap geodata. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1645–1650, Madeira Island, Portugal.

[Hido *et al.*, 2011] Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., and Kanamori, T. (2011). Statistical outlier detection using direct density ratio estimation. *Knowledge and Information Systems*, 26(2):309–336.

[Irie and Tomono, 2012] Irie, K. and Tomono, M. (2012). Localization and road boundary recognition in urban environments using digital street maps. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 4493–4499, St. Paul, Minnesota, USA.

[iRobot, 2016] iRobot (2016). iRobot roomba vacuuming robot. `http://www.irobot.com/For-the-Home/Vacuuming/Roomba.aspx`. [Accessed: 20-May- 2016].

[Jochem *et al.*, 1995] Jochem, T., Pomerleau, D., Kumar, B., and Armstrong, J. (1995). PANS: a portable navigation platform. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 107–0112.

[Joshi and James, 2015] Joshi, A. and James, M. R. (2015). Generation of accurate lane-level maps from coarse prior maps and lidar. *IEEE Intelligent Transportation Systems Magazine*, 7(1):19–29.

[Kelley, 1987] Kelley, C. T. (1987). *Iterative Methods for Optimization (Frontiers in Applied Mathematics).* Society for Industrial and Applied Mathematics.

[Kobayashi *et al.*, 2001] Kobayashi, H., Kikuchi, K., Ochi, K., and Onogi, Y. (2001). Navigation strategies referring to insect homing in flying robots. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, volume 2, pages 1695–1700.

[Kong *et al.*, 2010] Kong, H., Audibert, J., and Ponce, J. (2010). General road detection from a single image. *IEEE Transactions on Image Processing*, 19(8):2211–2220.

[Kümmerle *et al.*, 2013] Kümmerle, R., Ruhnke, M., Steder, B., Stachniss, C., and Burgard, W. (2013). A navigation system for robots operating in crowded urban environments. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 3225–3232, Karlsruhe, Germany.

[Kwak *et al.*, 2011] Kwak, K. H., Huber, D., Badino, H., and Kanade, T. (2011). Extrinsic calibration of a single line scanning lidar and a camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 3283–3289, San Francisco, USA.

[Langer and Thorpe, 1992] Langer, D. and Thorpe, C. (1992). Sonar based outdoor vehicle navigation and collision avoidance. In *Proceedings of* *the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 1445–1450, Raleigh, USA.

[Leonard and Durrant-Whyte, 1991] Leonard, J. and Durrant-Whyte, H. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382.

[Levinson and Thrun, 2010] Levinson, J. and Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 4372–4378, Anchorage, Alaska, USA.

[Levinson and Thrun, 2013] Levinson, J. and Thrun, S. (2013). Automatic online calibration of cameras and lasers. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany.

[Li and Olson, 2010] Li, Y. and Olson, E. B. (2010). A general purpose feature extractor for light detection and ranging data. *Sensors*, 10(11):10356.

[Lu *et al.*, 2014] Lu, W., Rodriguez F., S., Seignez, E., and Reynaud, R. (2014). Probabilistic error model for a lane marking based vehicle localization coupled to open source maps. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 360–365.

[Mastin *et al.*, 2009] Mastin, A., Kepner, J., and Fisher, J. (2009). Automatic registration of lidar and optical images of urban scenes. In *Proceedings of IEEE Computer Society Conference on Vision & Pattern Recognition (CVPR)*, pages 2639–2646.

[Matsushita and Miura, 2011] Matsushita, Y. and Miura, J. (2011). On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter. *Robotics and Autonomous Systems*, 59(5):274–284.

[Misawa *et al.*, 2007] Misawa, M., Yoshida, T., and Yuta, S. (2007). A smart handcard with autonomous returning function. *Journal of Robotics Society of Japan*, 25(8):1199–1206.

[Montemerlo *et al.*, 2008] Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., and Thrun, S. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597.

[Moon *et al.*, 1995] Moon, Y.-I., Rajagopalan, B., and Lall, U. (1995). Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3):2318–2321.

[Morales *et al.*, 2009] Morales, Y., Tsubouchi, T., and Yuta, S. (2009). Vehicle 3D localization in mountainous woodland environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 3588–3594, St. Louis, Missouri, USA.

[Napier *et al.*, 2013] Napier, A., Corke, P., and Newman, P. (2013). Cross-calibration of push-broom 2D lidars and cameras in natural scenes. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 3679–3684, Karlsruhe, Germany.

[Oommen *et al.*, 1987] Oommen, B., Iyengar, S., Rao, N., and Kashyap, R. (1987). Robot navigation in unknown terrains using learned visibility graphs. part i: The disjoint convex obstacle case. *IEEE Journal on Robotics and Automation*, 3(6):672–681.

[Pandey, 2015] Pandey, G. (2015). Extrinsic calibration of a 3d lidar and camera. `http://robots.engin.umich.edu/SoftwareData/ExtrinsicCalib`. [Accessed: 3- Mar- 2016].

[Pandey *et al.*, 2012] Pandey, G., McBride, J. R., Savarese, S., and Eustice, R. M. (2012). Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 2053–2059.

[Pandey *et al.*, 2014] Pandey, G., McBride, J. R., Savarese, S., and Eustice, R. M. (2014). Toward mutual information based place recognition. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 3185–3192, Hong Kong, China.

[Parra Alonso *et al.*, 2012] Parra Alonso, I., Fernandez Llorca, D., Gavilan, M., Alvarez Pardo, S., Garcia-Garrido, M., Vlacic, L., and Sotelo, M. (2012). Accurate global localization using visual odometry and digital maps on urban environments. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1535–1545.

[Peng *et al.*, 2005] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.

[Pethel and Hahs, 2014] Pethel, S. D. and Hahs, D. W. (2014). Exact test of independence using mutual information. *Entropy*, 16(5):2839–2849.

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

[Rusu and Cousins, 2011] Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, Shanghai, China.

[Sainui and Sugiyama, 2013] Sainui, J. and Sugiyama, M. (2013). Direct approximation of quadratic mutual information and its application to dependence-maximization clustering. *IEICE Transactions*, 96-D(10):2282–2285.

[Sakai and Sugiyama, 2014] Sakai, T. and Sugiyama, M. (2014). Computationally efficient estimation of squared-loss mutual information with multiplicative kernel models. *IEICE Transactions on Information and Systems*, E97-D(4):968–971.

[Scaramuzza *et al.*, 2007] Scaramuzza, D., Harati, A., and Siegwart, R. (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 & Systems (IROS)*, pages 4164–4169, San Diego, USA.

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

[Scott and Sain, 2005] Scott, D. W. and Sain, S. R. (2005). Multidimensional density estimation. In Dey, D. and Rao, C., editors, *Data Mining and Data Visualization, Handbook of Statistics*, chapter 9, pages 229–262. Elsevier.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

[Shaw, 2004] Shaw, J. R. (2004). QuickFill: An efficient flood fill algorithm. `http://www.codeproject.com/Articles/6017/QuickFill-An-efficient-flood-fill-algorithm`. [Accessed: 7-May- 2016].

[Suzuki *et al.*, 2009a] Suzuki, T., Sugiyama, M., Kanamori, T., and Sese, J. (2009a). Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(1):1–12.

[Suzuki *et al.*, 2009b] Suzuki, T., Sugiyama, M., and Tanaka, T. (2009b). Mutual information approximation via maximum likelihood estimation of density ratio. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pages 463–467, Seoul, Korea.

[Székely and Rizzo, 2009] Székely, G. J. and Rizzo, M. L. (2009). Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1236–1265.

[Székely *et al.*, 2007] Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794.

[Tangkaratt *et al.*, 2015] Tangkaratt, V., Xie, N., and Sugiyama, M. (2015). Conditional density estimation with dimensionality reduction via squared-loss conditional entropy minimization. *Neural Computation*, 27(1):228–254.

[Tao *et al.*, 2013] Tao, Z., Bonnifait, P., Fremont, V., and Ibanez-Guzman, J. (2013). Mapping and localization using gps, lane markings and proprioceptive sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 406–412, Tokyo, Japan.

[Taylor and Nieto, 2012] Taylor, Z. and Nieto, J. (2012). A mutual information approach to automatic calibration of camera and lidar in natural environments. In *Proceedings of The Australian Conference on Robotics and Automation (ACRA)* , pages 3–5, Wellington, New Zealand.

[Taylor *et al.*, 2013] Taylor, Z., Nieto, J., and Johnson, D. (2013). Automatic calibration of multi-modal sensor systems using a gradient orientation measure. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 1293–1300, Tokyo, Japan.

[The MathWorks, Inc., 2015] The MathWorks, Inc. (2015). Find minimum of unconstrained multivariable function - MATLAB fminunc. `http://www.mathworks.com/help/optim/ug/fminunc.html`. [Accessed: 19- Mar-2016].

[Thorpe *et al.*, 1988] Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. (1988). Vision and navigation for the carnegie-mellon navlab. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 10(3):362–373.

[Thrun, 2010] Thrun, S. (2010). Official google blog: What we're driving at. `https://googleblog.blogspot.jp/2010/10/what-were-driving-at.html`. [Accessed: 15- Jun- 2016].

[Thrun *et al.*, 1999] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). MINERVA: a second-generation museum tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, volume 3, pages 1999–2005.

[Thrun *et al.*, 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

[Thrun *et al.*, 2000] Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2000). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.

[Tomono, 2009] Tomono, M. (2009). Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 4306–4311.

[Torkkola, 2003] Torkkola, K. (2003). Feature extraction by non parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438.

[Wangsiripitak and Murray, 2009] Wangsiripitak, S. and Murray, D. W. (2009). Avoiding moving outliers in visual slam by tracking moving objects. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 375–380.

[Wells *et al.*, 1996] Wells, III, W. M., Viola, P., Atsumi, H., Nakajima, S., and Kikinis, R. (1996). Multi-modal volume registration by maximization of mutual information. *Multi-modal volume registration by maximization of mutual information*, 1(1):35–51.

[Williams *et al.*, 2004] Williams, N., Low, K.-L., Hantak, C., Pollefeys, M., and Lastra, A. (2004). Automatic image alignment for 3d environment modeling. In *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing*, pages 388–395.

[Yoshida *et al.*, 2010] Yoshida, T., Irie, K., Koyanagi, E., and Tomono, M. (2010). A sensor platform for outdoor navigation using gyro-assisted odometry and roundly-swinging 3D laser scanner. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 1414–1420.

[Zennaro *et al.*, 2015] Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., and Menegatti, E. (2015). Performance evaluation of the 1st and 2nd generation kinect for multimedia applications. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[Zhang and Pless, 2004] Zhang, Q. and Pless, R. (2004). Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems (IROS)*, pages 2301–2306.