

# Isovist-induced Robust LiDAR Localization

Giseop Kim

Advisor: Ayoung Kim

A dissertation submitted to the faculty of  
Korea Advanced Institute of Science and Technology in  
partial fulfillment of the requirements for the degree of  
Master of Science in Civil and Environmental Engineering

Daejeon, Korea  
December 20, 2019

Approved by

---

Ayoung Kim  
Professor of Civil and Environmental Engineering

The study was conducted in accordance with Code of Research Ethics<sup>1</sup>.

---

<sup>1</sup> Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

## **Abstract**

For the era of autonomous cars, the accurate and reliable positioning of a vehicle is critical. However, a city is still not easy for them. For example, tall buildings disrupt the GNSS signal. Therefore, it is necessary for a robot to estimate the position using only the surrounding information obtained from equipped sensors. However, the appearance of a place is diverse. Day and night are different. Dynamic objects appear and disappear. A building that existed yesterday could be demolished today. In this thesis, we explore the intrinsic feature of a place that distinguishes that place from others.

How does a human recognize a place? In the field of urban design, there has been a concept called *isovist*. The isovist is an observer's egocentric visibility and means the openness of a space. The openness that an observer feels in the space also determines the use of that space. For example, in a square, we get the feeling that we are open and that it is closed between high-rise buildings. The openness of the space refers to how robust it is in the presence of dynamic objects and light condition changes.

This thesis proposes a robust robot localization method using a LiDAR. Because light goes straight, the shape of the surrounding environment obtained from a LiDAR is the robot's egocentric visible space's shape. Using this point cloud, the data-driven three-dimensional (3D) isovist is proposed and employed for robot localization. That is, in this thesis, robot localization meets 3D isovist. Extensive experiments are conducted to cover diverse environments and times, and the results—for example, those related to placeness—might come from the openness.

**Keywords** Mobile robot, Localization, Light Detection And Ranging (LiDAR), Point cloud, 3D isovist

# Contents

Contents . . . . .	i
List of Tables . . . . .	iii
List of Figures . . . . .	iv
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Thesis Overview . . . . .	4
1.3 Related Publications and Presentations . . . . .	5
<b>Chapter 2. Background</b>	<b>6</b>
2.1 Robot Localization . . . . .	6
2.1.1 Taxonomy of Robot Localization . . . . .	6
2.1.2 Problem 1: Place Recognition for SLAM . . . . .	9
2.1.3 Problem 2: Long-term Localization . . . . .	11
2.2 LiDAR Localization: Literature Review . . . . .	12
<b>Chapter 3. Scan Context: 3D Isovist for Robot Localization</b>	<b>15</b>
3.1 Isovist . . . . .	15
3.1.1 Introduction . . . . .	15
3.1.2 Sensor data-driven 3D Isovist . . . . .	17
3.2 Scan Context (SC): Egocentric Place Descriptor . . . . .	20
<b>Chapter 4. Application 1: Online Place Recognition</b>	<b>25</b>
4.1 Introduction . . . . .	25
4.2 Related Work . . . . .	26
4.3 Scan Context for Place Recognition . . . . .	27
4.3.1 Similarity Score between Scan Contexts . . . . .	28
4.3.2 Two-phase Search Algorithm . . . . .	29
4.4 Experimental Evaluation . . . . .	30
4.4.1 Dataset and Experimental Settings . . . . .	30
4.4.2 Precision Recall Evaluation . . . . .	32
4.4.3 Localization Accuracy . . . . .	34
4.4.4 Computational Complexity . . . . .	34
4.5 Conclusion . . . . .	34

<b>Chapter 5.</b>	<b>Application 2: Long-term Localization</b>	<b>36</b>
5.1	Introduction . . . . .	36
5.2	SCI Generation and Training . . . . .	38
5.2.1	A brief review of Scan Context (SC) . . . . .	39
5.2.2	Scan Context Image (SCI) . . . . .	39
5.2.3	Location Definition . . . . .	39
5.2.4	Network Selection . . . . .	39
5.2.5	N-way SCI Augmentation . . . . .	40
5.3	SCI Localization . . . . .	42
5.3.1	Un-learned Place Detection . . . . .	42
5.3.2	Localization . . . . .	43
5.4	Experiments . . . . .	43
5.4.1	Benchmark Datasets . . . . .	43
5.4.2	Comparison Methods . . . . .	45
5.5	Evaluation Results . . . . .	45
5.5.1	Precision-recall Curve . . . . .	45
5.5.2	Retrieval Capability . . . . .	46
5.5.3	Long-term Robustness . . . . .	46
5.5.4	Robustness to Viewpoint Changes . . . . .	47
5.5.5	Grid Cell Size . . . . .	48
5.5.6	Runtime Evaluation . . . . .	49
5.6	Conclusion . . . . .	50
<b>Chapter 6.</b>	<b>Conclusion</b>	<b>51</b>
6.1	Contributions . . . . .	51
6.2	Future Work . . . . .	51
	<b>Bibliography</b>	<b>52</b>
	<b>Acknowledgments in Korean</b>	<b>59</b>

## List of Tables

4.1	Selected dataset lists used in validation . . . . .	31
4.2	Average time costs on KITTI00. . . . .	34
5.1	The structure of the classification network we used . . . . .	42
5.2	Summary of two long-term dataset: NCLT and Oxford RobotCar . . . . .	45
5.3	Average time cost for each methods. The comparison is conducted on the 2013-04-05 of the NCLT dataset. . . . .	50

## List of Figures

1.1	Autonomous robots usually have multimodal sensors such as camera and LiDAR for the intelligent perception. (a) The Segway robotic platform with multiple sensors (Image courtesy of Nicholas Carlevaris-Bianco [10]). (b) LiDAR sensor system for the complex urban data set (Image courtesy of Jinyong Jeong [31]). . . . .	1
1.2	Comparison of an image and a point cloud acquired at a same place . . . . .	3
2.1	Taxonomy of robot localization . . . . .	7
2.2	An example of metric localization and coarse localization. . . . .	8
2.3	Importance of Loop Closure Detection (LCD) and loop closing . . . . .	9
2.4	An real world example of motion drift and pose optimization . . . . .	10
2.5	Example captures from Complex Urban Dataset [31]. With the advancement of SLAM technology, it is now easier to get a large scale point cloud map like this, which allows non-mapper robots to focus on localization. . . . .	11
2.6	The robustness of structural data . . . . .	12
2.7	Taxonomy of robot localization methods using point cloud from LiDAR sensors . . . . .	14
3.1	Example of isovists . . . . .	15
3.2	Limitation of 2D isovist . . . . .	16
3.3	An isovist polygon of a line of sight . . . . .	17
3.4	Example of 3D isovist on a Digital Elevation Model (DEM) . . . . .	17
3.5	Visualization of a LiDAR scan . . . . .	18
3.6	Example of sensor data-driven 3D isovist . . . . .	19
3.7	Advantages of sensor data-driven 3D isovist . . . . .	19
3.8	Preview of Scan Context . . . . .	20
3.9	Scan Context creation . . . . .	21
3.10	An example of Scan Context . . . . .	23
3.11	Example of scan contexts from the same place with time interval . . . . .	24
4.1	Overview of Scan Context-based place recognition algorithm . . . . .	26
4.2	Example captures from Complex Urban Dataset [31]. With the advancement of SLAM technology, it is now easier to get a large scale point cloud map like this, which allows non-mapper robots to focus on localization. . . . .	28
4.3	The ring key generation for the fast search. . . . .	30
4.4	Selected datasets for the evaluation . . . . .	31
4.5	Precision-recall curves for the evaluation datasets. . . . .	32
4.6	A challenging example captured from Complex Urban LiDAR dataset sequence 02. . . . .	33
4.7	Computation time and RMSE with and without Scan Context. . . . .	34
4.8	An example of point-to-point ICP results using Scan Context or not from KITTI 08. . . . .	35
5.1	The concept of our 1-day learning and 1-year localization . . . . .	36
5.2	Overall pipeline of the SCI localization and performance evaluation. . . . .	37

5.3	Scan Context Image (SCI) generation process . . . . .	40
5.4	Examples of gridded map of NCLT and Oxford RobotCar datasets . . . . .	41
5.5	The example of the distribution of entropies of prediction score vectors for seen and unseen. . . . .	43
5.6	Visualization of point clouds and the associated SCIs for 3D and 2D LiDAR. . . . .	44
5.7	Precision-recall curves for two long-term datasets, NCLT and Oxford RobotCar dataset. . . . .	46
5.8	AUC performance changes over time for different criteria of success localization. . . . .	47
5.9	Robustness to non-structural changes. Despite challenging factors (e.g., viewpoint changes, occlusions, and foliage), the proposed method successfully found its location with a high score for over hundreds of places over a year. . . . .	48
5.10	Robustness to structural changes on the test sequences of the NCLT dataset. . . . .	48
5.11	Robustness to random viewpoint changes on the test sequences of the NCLT dataset. Each line is a mean for 10 test sequences. . . . .	49
5.12	Performances with respect to different grid cell sizes. . . . .	49

# Chapter 1. Introduction

## 1.1 Motivation

The era of self-driving cars is coming. Weimo, a subsidiary of the autonomous drive of the Google alphabet, has achieved mileage of 10 million miles (16,000,000 km) in November 2018. It is only 15 years since the Defense Advanced Research Projects Agency (DARPA) held its first autonomous driving car competition in 2004. The car that ran the longest distance at that time only ran 11.9 kilometers. Now the car is no longer car. It is an autonomous robot. SLAM technology, which has been studied for the last 20 years since the seminal works from Smith and Cheeseman [64] and Durrant-Whyte [18], and deep learning, which has developed explosively since AlexNet [39], 2012, now makes the autonomous vehicles.

The autonomous robot needs sense organs for intelligence navigation. A camera is the most popular sensor, and Radar and Light Detection and Ranging (LiDAR) are also common in the autonomous driving. Fig. 1.1 shows examples of autonomous robots equipped with multiple sensors.

Autonomous vehicles or autonomous robots perform missions (e.g., navigation for an autonomous taxi at urban sites, a robot at rescue sites, and self-exploration) by fusing various sensor information. Particularly, a mobile robot is required to first know their position accurately in order to perform mission. This the first essential requirement is called *localization*. This thesis explores how autonomous robots perform accurate localization in the city. However, it is not easy for the autonomous robot to operate in a city. Because of the presence of high-rise buildings (so called urban canyon), the error of the Global Positioning System (GPS) signal is large. Therefore, it is important to reliably localize by using the robot's own sensor rather than GPS.

Camera is commonly used robot localization because a camera is cheaper than other sensors. This



(a) The Segway robotic platform



(b) LiDAR sensor system

Figure 1.1: Autonomous robots usually have multimodal sensors such as camera and LiDAR for the intelligent perception. (a) The Segway robotic platform with multiple sensors (Image courtesy of Nicholas Carlevaris-Bianco [10]). (b) LiDAR sensor system for the complex urban data set (Image courtesy of Jinyong Jeong [31]).



type of method is called visual localization [21, 44]. However, an image acquired from a same place could be visually different as in Fig. 2.6(a), which can occur false localization, because camera is vulnerable to light condition changes. Therefore, a LiDAR sensor is commonly used to compensate the shortcomings of the camera. The LiDAR sensor recognizes the light reflected back from the object and calculates the distance between the object and the robot. Therefore, the robot can accurately capture the shape of the surrounding scene structure. This has the advantage of being able to capture the canonical features of the scene without regard to time and season, since it is not affected by light conditions as in Fig. 2.6(b).

However, the fact that a point cloud obtained from a LiDAR is *unstructured* unlike the image obtained from the camera makes it difficult to use the LiDAR sensor for robot localization. Unstructured data means that the shape of the data is not known in advance, unlike an image in which pixels are arranged. Fig. 1.2 shows the example. The camera produces an image of a fixed shape (i.e., non-changing), regardless of location. For example, images, the left column of Fig. 1.2 have 1232 by 1616 pixels. However, in the case of point cloud data, it is hard to predict how many points will be acquired unlike image. For example, point clouds, the right column of Fig. 1.2 have the different number of points and their points are distributed in the different range, which is not known in advance. Also, the point cloud may not be well preserved in downsample because neighbor relationship is not clear unlike pixel in image. For these characteristics we say the point cloud unstructured unlike image.

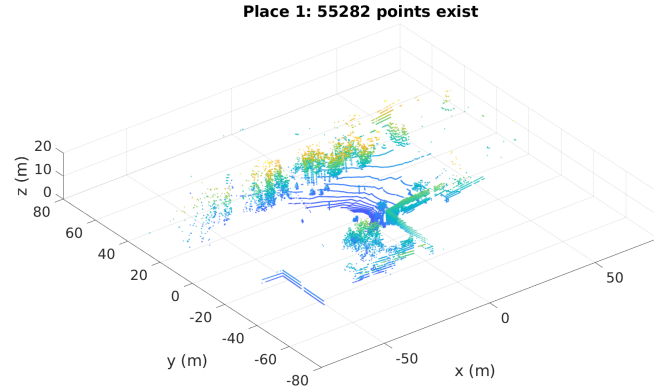
This thesis propose a novel method to summarize the unstructured point cloud for robot localization. Meanwhile, with the development of simultaneous localization and mapping (SLAM), a large scale of map (i.e., information about a place in a city-scale) has been constructed easily [31] nowadays. Thus using a pre-built map, the robot can do better localization [30, 75]. However, in order to make it possible, the place must be summarized effectively and efficiently. *Effectively* means that the summarized information of the place should be sufficiently distinguishable from other places. *Efficiently* means the summarized data should be lightweight. This thesis proposes a novel method for robot localization using LiDAR sensor and LiDAR point cloud map. Our contributions are:

- A novel point cloud descriptor. Conventional methods to describe a point clouds summarize the statistical properties of the point cloud using histograms (e.g, a histogram of normal vectors of points) and made a vector of a certain length. A method has been recently proposed for the network to generate a vector of defined length by directly consume the constant number of points. These existing methods mainly summarize the point cloud as a 1D vector. Therefore, when searching for the nearest place from the database, a naive matching method was applied (e.g., Euclidean norm between two vectors). However, unlike the existing method, we summarize the point cloud in 2D matrix form and propose a novel matching algorithm. This novel representation preserves the original shape of the point cloud and thus has a robustness than the other methods.
- A novel nearest point cloud retrieval algorithm. We propose two new retrieval algorithms for our novel point cloud descriptor. One is suitable for the problem of online place recognition and the other is for long-term localization. Details are presented in chapters §4 and §5, respectively.
- Viewpoint invariant. A robot sometimes revisits the same place in a way that it has not experienced before (e.g., reverse revisit). In this case, however, the robot should recognize the place despite the change in viewpoint. Unlike other methods, we align the viewpoints of two point clouds by performing coarse yaw registration in the matching process. This makes it possible to perform robust localization against viewpoint changes.

- Long-term robustness. Another important requirement for robots is long-term autonomy [41]. However, a robot hardly knows the all appearances (e.g., from day and night) of the place in advance. The robot should be able to recognize the place robustly over a long-term with only a few



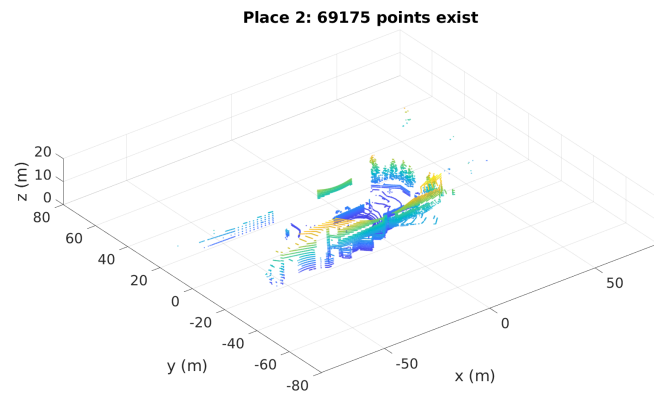
(a) Image (Place 1)



(b) Point cloud (Place 1)



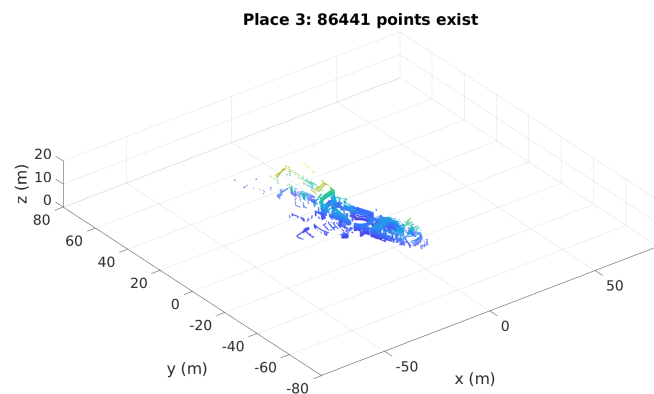
(c) Image (Place 2)



(d) Point cloud (Place 2)



(e) Image (Place 3)



(f) Point cloud (Place 3)

Figure 1.2: An example of point cloud's unstructuredness. Unlike images that have the same shape even if they are acquired at different places, the point clouds vary in size and range (and those are not known in advance).

experiences (or even a single experience). The retrieval algorithm proposed with our representation outperforms existing methods for this long-term localization problem.

## 1.2 Thesis Overview

In this section, the outline of this thesis is provided.

### Chapter 1

In this chapter, we illuminate our two problems to be solved and summarize the contributions of this thesis in these two problem areas.

### Chapter 2

This chapter explains why the two problems we target to solve are important. The first problem is **online place recognition** for SLAM. Place recognition (or loop closure detection) is an essential process for a robot to make a globally consistent map. The second problem is **long-term localization**. Apart from the robot that creates the map through the SLAM algorithm, other robots can localize using that map created by the mapper robot. Therefore, the robot can focus only on localizing and obtain better localization results with the same computing resources.

### Chapter 3

Our main contribution, the invention of *Scan Context*, and its details are introduced in this chapter. Scan Context is a novel point cloud descriptor, which is induced from the concept of urban visibility and 3D Isovist. Therefore we first introduce the concept of isovist, which has been widely used in urban design. The definition of Scan Context and several interesting characteristics of Scan Context are then described. Finally, we examine the meaning of Scan Context in both mobile robotics and urban design.

### Chapter 4

Our first target problem, online place recognition, is introduced. A novel pipeline using Scan Context for place recognition is proposed and comparisons with existing histogram-based point cloud descriptors are provided for various datasets in various environments (e.g., from a campus to a metropolitan).

### Chapter 5

Our second target problem, long-term localization, is introduced. Scan Context is reformulated into the 3-channel normalized data, which is called Scan Context Image. This image-format data is fed to a convolutional neural network (CNN) and we propose a novel long-term localization pipeline using this prediction vectors from a CNN.

### Chapter 6

The overall conclusion and future works are discussed.

### 1.3 Related Publications and Presentations

- Giseop Kim, Hyunchul Roh, Youngchul Kim, and Ayoung Kim, Sensor Data-driven Urban Site Analysis using Point Cloud from Urban Mapping System, Late Breaking Results at ICRA, Singapore, May, 2017
- Giseop Kim, Byungjae Park and Ayoung Kim, Learning Scan Context toward Long-term LiDAR Localization. In ICRA Workshop on Long-term Autonomy and Deployment of Intelligent Robots in the Real-world, Brisbane, May. 2018. (Best paper award).
- Giseop Kim and Ayoung Kim, Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October. 2018
- Giseop Kim, Byungjae Park and Ayoung Kim, 1 Day Learning, 1 Year Localization: Long-term LiDAR Localization using Scan Context Image, IEEE Robotics and Automation Letters (with ICRA), 2019. (Under review. Revised and resubmitted)
- Giseop Kim, Ayoung Kim and Youngchul Kim, A new 3D space syntax metric based on 3D isovist capture in urban space using remote sensing technology, Computers Environment And Urban System, 2019.

## Chapter 2. Background

### 2.1 Robot Localization

The word, *robot localization*, means any algorithm make a robot estimate its positional information, which is also called simply ‘pose’. The pose can be defined in any mathematical format depending on the purpose of the application. Global Navigation Satellite System (GNSS) systems, for example, represent locations in terms of latitude, longitude, and altitude. Also, the position of the robot on the 2D plane can be expressed as  $x$  and  $y$ . Localization is the essential task that should be performed first for an autonomous robot to achieve a mission in next levels such as path planning, navigation, or manipulation.

In the 21st century, GNSS have been supplied to most cars, and GNSS became the most basic and easy sensor for robot to localize. However, since the GNSS signal is unreliable in a city (e.g., a few or a few tens of meters error), there have been many studies on robot localization using robot’s own sensors such as camera or LiDAR.

In this section, first, the concept and several categories of robot localization is described. Our main target problem is global coarse localization, which is introduced in the following paragraph, among the several types of localization. The overall taxonomy is illustrated in Fig. 2.1.

Second, in subsection §2.1.2, we explain why we need to solve that problem in terms of SLAM. Then, in the following subsection §2.1.3, we also explain why robots need global coarse localization to focus more on localization without mapping.

#### 2.1.1 Taxonomy of Robot Localization

Robot localization refers finding current position information of robot. Depending on how position information is estimated, robot localization can be divided into two categories; *tracking* and *global localization*.

**Tracking** literally tracks pose information based on previous information. Tracking can be divided into two type of methods, one using map [73] and one not using map (usually called odometry). For odometry category, in recent years, there have been many methods of calculating the relative pose transformation between two frames using only a single camera [19, 52]; thus this is usually called visual odometry using mono camera. In recent years, with the development of this visual odometry, a robot has an error of only a few centimeters while moving a few tens of meters. The more in-depth introduction of visual odometry is can be found at these papers [20, 60]

**Global localization** directly estimates a robot pose within a global frame, not the relative transformation. Also, this method sometimes aims to use only current sensor measurements without using the previous information. Global localization is required for the following reasons.

1. Drift correction.

As mentioned above, a robot iteratively estimates its position through odometry via calculating a relative transformation between a current and a previous sensor measurement. However, since sensors (e.g., camera and LiDAR) of a robot has a natural noise, a small error of relative position exists and the error is accumulated according to the motion of the robot as in Fig. 2.3(a) and Fig. 2.3(b). Therefore, as shown in Fig. 2.3(b), even if the robot returns to the same place, there is

a difference between the robot’s expected position and its actual position. To reduce this unwanted error, we need to add constraints between these two poses and perform pose-graph optimization. This process is called loop closure and is described in more detail in section §2.1.2. When the robot revisits the place, we call the drift correction ‘close the loop’ because the optimized trajectory’s shape forms a loop. However, in order to perform loop closing, the process of loop detection (revisit detection) should be preceded. Therefore, global localization plays a role of loop detection for drift correction.

## 2. Relocalization

Sometimes the robot needs to be globally aware of the position even though it is not a loop (revisit) condition. For example, tracking sometimes fails. Tracking, which relies on relative information, may not predict the relative motion correctly if the quality of current or previous information is not good and this phenomenon affects all subsequent predictions. Therefore, when the tracking fails, it is necessary to re-estimate the current position globally against the previous positions. This process is called relocalization and is one of the reasons a robot need global localization.

## 3. Initialization

Although it is possible to use the coordinate of start point of the robot as the origin, in order to operate the robot in the real world, it is necessary to know the current position in relation to existing objects and buildings. In other words, global coordinates are required from the beginning for robot navigation or mission. Therefore, global localization is required because the robot needs to know the position of the global coordinates before tracking.

## 4. Separation of Mapper and Localizer

Global localization is also necessary for robots that only perform localization. Using SLAM, which is briefly introduced in the following section §2.1.2, a robot can estimate its pose without prior knowledge of the environment. However, in order to estimate a precise position and to drive a

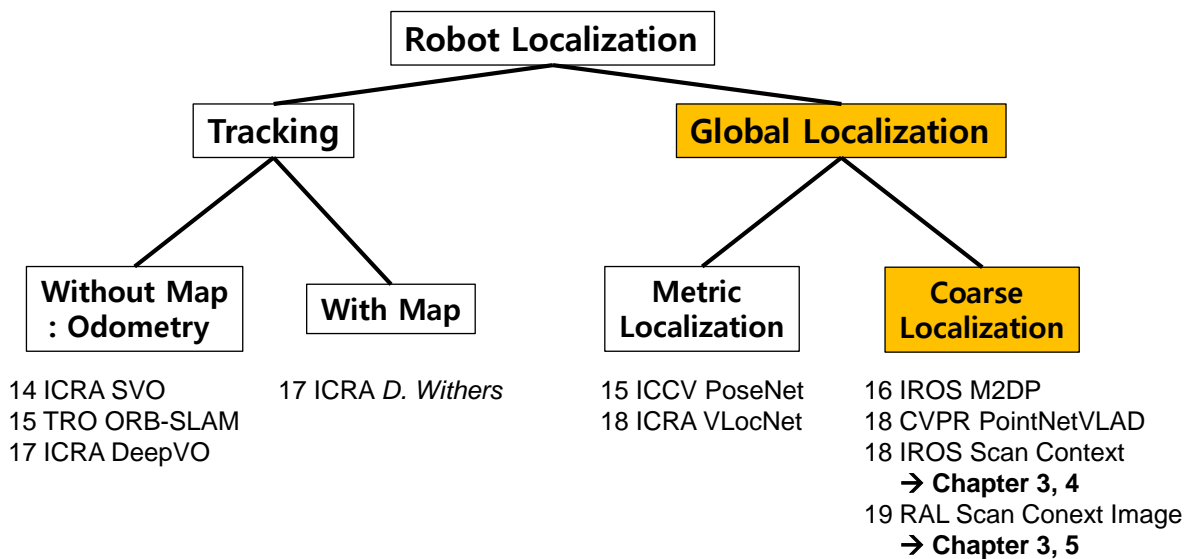
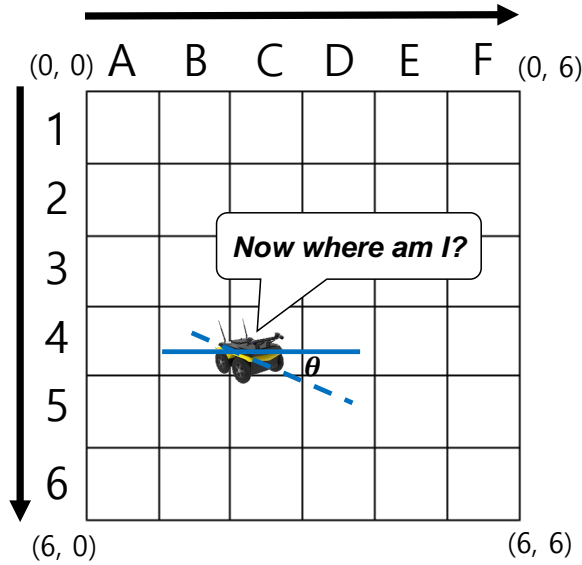


Figure 2.1: Taxonomy of robot localization. Depending on whether localization is performed using previous pose information, it is divided into tracking and global localization.



- **Metric Localization**
  - “I am at  $(x = 3.7, y = 2.2, \text{heading}(\theta) = -45^\circ)$ ”
- **Coarse Localization**
  - “I am at C4”

Figure 2.2: An example of metric localization and coarse localization.

robot in real world (e.g., autonomous vehicle in a city), a robot should use a pre-built map (e.g., 3D point cloud, but any type of information of scene can be said ‘map’) constructed by another robot in advance. Using the pre-built map, the robot can localize globally (i.e., direct inference of current position without tracking). Therefore, it is possible to separate the robot that generates the map and the robot that performs only the localization. With global localization techniques using the map, the robot, which has limited computing power, could concentrate only on the localization.

1 and 2 of aforementioned four reasons are needed for SLAM. A formal description of SLAM and why global localization is important for SLAM is discussed in more detail in section §2.1.2. 3 and 4 are needed for robot to robustly localize in a real-world and details are given in section §2.1.3.

Global localization is again divided into two subcategories; *metric* and *coarse*. **Metric global localization** aims to predict a precise (practically under a few centimeter-level) continuous real value of 3DoF ( $x$ ,  $y$ , and heading in 2D environment) or 6DoF (3 for location and 3 for rotation) state of robot within a global coordinate. Differently, **Coarse global localization** provides a rough scope (but practically precision within a few meters is required) of current location of the robot. The output of metric localization is in continuous space and the output of coarse localization is discrete. A simple but effective example is given in Fig. 2.2.

Metric localization is used for AR or VR because it provides centimeter-level precise positioning. However, since it is more complicated than coarse localization, it is still difficult to perform for a wide range (i.e., hard to be scalable). For example, in the case of the 7-Scenes RGB-D Dataset [63], which is widely used for global metric localization researches, the coverage of the motion is just a few square meters. On the other hand, the size of a city, which is an environment where autonomous robots such as unmanned vehicles operate, reaches a few square kilometers. For example, a length of robot motion

from Oxford RobotCar dataset [47] is nearly over 10 km. Coarse localization is first required for loop detection as mentioned already. Also, it is proper for scalable operation of robot.

The overall summarization of the taxonomy of robot localization is visualized in Fig. 2.1. We note that this thesis’s target problem belongs to coarse global localization.

### 2.1.2 Problem 1: Place Recognition for SLAM

We again note that the target problem of this thesis is global coarse localization and, in the previous section, we introduced the fact that one of the important roles of global coarse localization is loop closure detection (or called online place recognition for SLAM). In this section, we explore a necessity of loop closure, which is roughly mentioned in the previous chapter.

The term, SLAM, stands for **S**imultaneous **L**ocalization **A**nd **M**apping. SLAM is a method of estimating a robot pose while simultaneously making a map. If the robot experiences the environment for the first time, the robot does not have knowledge about the environment. Therefore, it organizes the

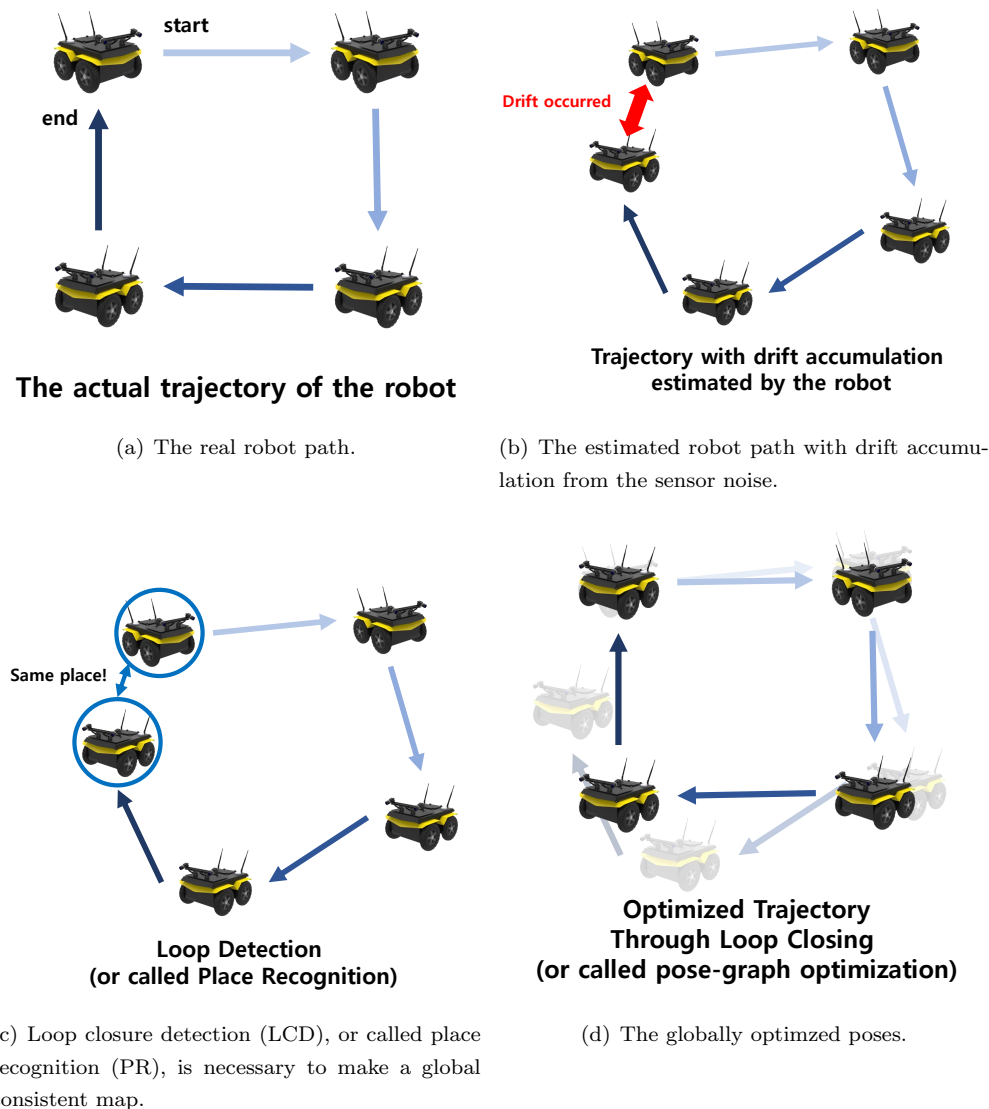
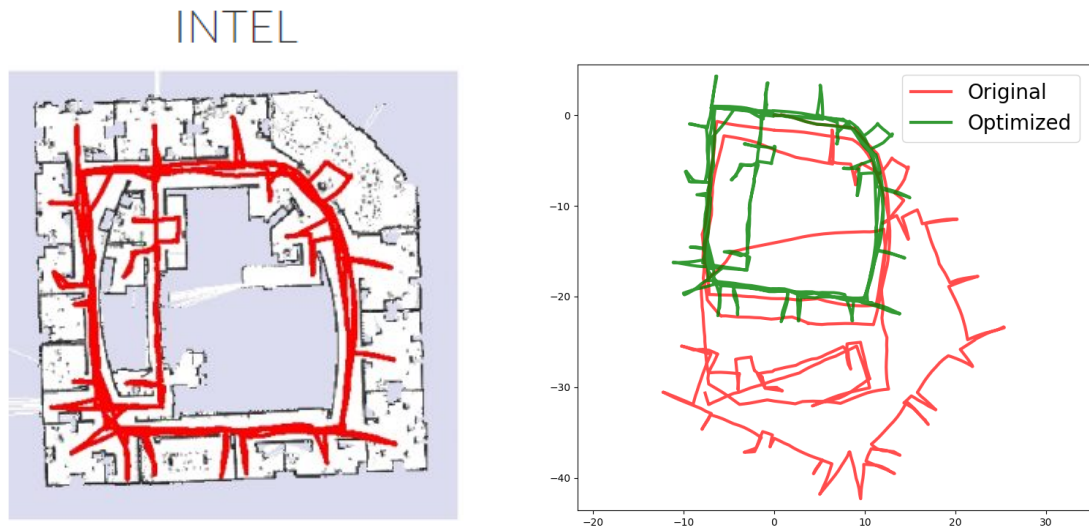


Figure 2.3: Importance of Loop Closure Detection (LCD) and loop closing





(a) The robot trajectory of the INTEL (Intel Research Lab data) overlaid on the building plan map. Note that the image is from <http://lucacarlone.mit.edu/datasets/>

(b) The original trajectory (green) and the optimized trajectory (blue) using Ceres solver [1]

Figure 2.4: An real world example of motion drift and pose optimization

information about the environment using only its own sensors and estimates its position based on this information (map). Map is quite abstract term and can be understood as an any type of information about the robot’s surrounding environment. Only a few type of map (e.g., occupancy map, 3D point cloud or set of images), however, is mainly used. In recent years, with help of the development of deep learning, a neural network containing knowledge about the environment is considered as a map. This thesis does not cover the whole SLAM, but only localization, therefore here is a good references to finish and replace the SLAM description here. A more in-depth history and theory of SLAM are found in these paper [4, 9, 17] and theses [33, 40, 54, 65, 67].

From the viewpoint of state estimation, SLAM can be conceptually defined as follows.

$$\text{SLAM} = \text{Odometry} + \text{Loop Closure}$$

Odometry predicts a relative motion between a previous state and a current motion. This relative motion is estimated via many sensors, for example, camera (visual odometry), wheel encoder (wheel odometry), or inertial measurement unit (IMU). However, the error of estimated robot motion is accumulated like Fig. 2.3(b) since those sensors have intrinsic noise or external noise (e.g., harsh light condition and rough terrain). Thus, robot need to reduce the unwanted error via graph optimization via representing discrete robot pose as a node and the connection between nodes as edge. Therefore, in order to perform the optimization, the connection between the two nodes obtained at different times in the revisit place should be added as a constraint for the optimization problem. This process is called a loop closure and the process of determining whether a current place is the revisited place that should be performed before loop closure optimization is called loop closure detection (Fig. 2.3(c)).

In short, estimation of robot pose and construction of globally consistent map, SLAM, is performed with two modules; the motion is basically estimated using the odometry, and the accumulated error is reduced through the loop closure and finally robot can get a global-consistent map. Fig. 2.4 shows a

real-world example of Fig. 2.3. As seen in Fig. 2.4(b), we know the motion estimation without loop closure (pose-graph optimization) is inaccurate.

Therefore, correct loop detection is required for successful pose optimization. The optimization may perform in the wrong way, which occurs wrong motion estimation or fail if the detected loop is actually not a loop (i.e., a current place is actually not a revisited place) so that false constraint is added. Therefore, a robust loop detector should not only have high detection performance but also minimize the case of misjudgment. Since the many SLAM researches have been mainly used cameras, many existing loop detectors have also been studied for images. Therefore, robust loop detectors for LiDAR sensors are rare compared to a camera. The first goal of this thesis is to develop an efficient and effective loop detector for LiDAR, which will be discussed in more detail in chapters 3 and 4.

### 2.1.3 Problem 2: Long-term Localization

In this section, we discuss the second key-role of global coarse localization and this is the second contribution point of this thesis, which will be guided in detail in section §5.

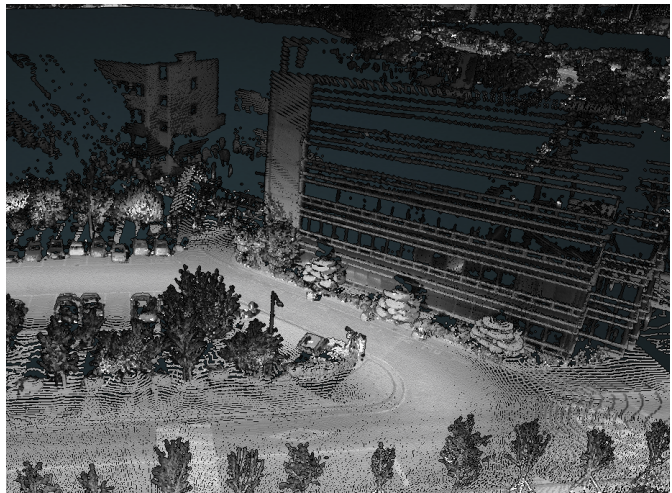
In the previous section §2.1.1, we discussed a robot could localize well and lessen the computing burden of making map if the pre-built map (i.e., any type of information about the environment) is available. In addition, coordinates in a global frame are necessary for the robot to interact with other robots in the real world rather than the self-centered coordinates. Such global coordinates are defined with respect to the pre-built map. The robot estimates the current global position by comparing the global map information with the information obtained from its own sensors.

In particular, when the robot visits a place previously experienced and has the map about the environments, the robot can effectively estimate the global coordinates by using the place information experienced in the past. Fig. 2.5 is an example of 3D point cloud map from Complex Urban LiDAR dataset [31].

This task is especially called **long-term localization** if the time difference between mapping and



(a) Top view of KAIST point cloud map.



(b) Near view of a building of a department of Civil and Environmental Engg.

Figure 2.5: Example captures from Complex Urban Dataset [31]. With the advancement of SLAM technology, it is now easier to get a large scale point cloud map like this, which allows non-mapper robots to focus on localization.

localization is large (e.g., day to night, summer to winter) so that environmental conditions are partially changed (e.g., light condition, color of appearance scene, or structural changes such as foliage falling or construction of a new building).

In order to localize robots robustly in the outdoor environment, not a static lab-like indoor environment, it is necessary to develop an method, which is invariant to the environmental changes over time. Unlike the SLAM technology in the indoor static environment is now commercialized-level and applied to a cleaner robot, research on the long-term autonomy is now in the beginning stage. SLAM researchers have also recognized the importance of this topic and have recently been engaged in a vigorous study such as holding a related workshop at a major robotics conference <sup>1</sup> and releasing long-term open datasets [10, 47].

This thesis argues that LiDAR is more effective than camera for long-term robot localization problem and details and related works are introduced in chapter §5.

## 2.2 LiDAR Localization: Literature Review

### Why LiDAR

We said this thesis solves global coarse localization problems such as 1. place recognition for SLAM and 2. robust long-term localization. Specifically, we propose ways to solve such problems using **LiDAR** sensors. LiDAR is one of the most predominant sensors supporting the perception of autonomous robots as camera. Since LiDAR sensor is used for various applications such as odometry and mapping [80], SLAM [28], object detection [45], intent prediction [11], and sensor fusion [62, 78], it is considered to be an essential sensor to be equipped with autonomous vehicles.

In order to solve the two problems we mentioned earlier in section §2.1.2 and section §2.1.3, LiDAR is more effective than the camera. Fig. 2.6 shows the reason. Camera is vulnerable to light conditions (Fig. 2.6(a)) and the appearance (i.e., color, texture, or shape such as edges) difference between day and night is large so that false loop detection may occur. A LiDAR sensor, on the other hand, captures structural information precisely (within a few centimeters), so measurements from LiDAR is robust to time changes and proper for both robust place recognition and long-term localization.

### Design Criteria for point cloud descriptor

Nevertheless, robust localization methods using LiDAR sensors have not much been studied than cameras. There are two reasons. First, a normal (3D) LiDAR is expensive than a camera. However,

<sup>1</sup><https://sites.google.com/view/icra2018ltaws/home>

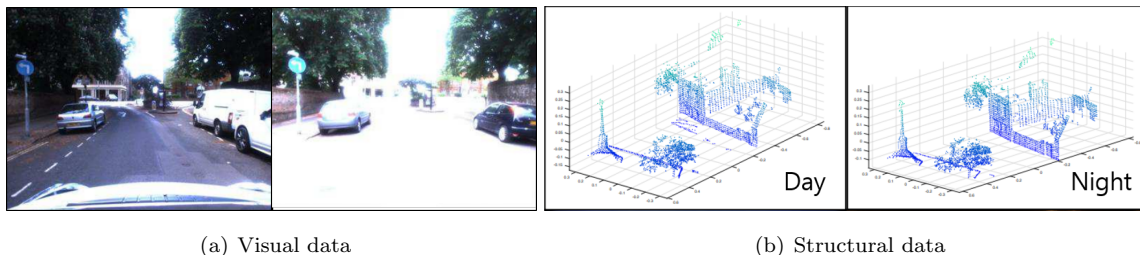


Figure 2.6: Note that the image Fig. 2.6(a) is from Oxford RobotCar dataset [47] and Fig. 2.6(b) is from PointNetVLAD [70].

due to the development of technology, LiDAR sensor price is getting lower. The second reason is more practical. Point cloud data from LiDAR is hundreds of thousands of points per second. Also, point cloud is an unstructured data; we can not know in advance a space a point cloud will occur, unlike an image, which has aligned pixels and downsampling to summarize large amounts of data can not guarantee preserving the geometric shape unlike images. Because of these difficulties, it is hard to extract and compress features from point clouds effectively than image. Moreover, in the outdoor environment we target, the LiDAR point cloud is noisy and the point cloud becomes more sparse as the distance from the robot increases. These facts makes it difficult to capture meaningful features from the point cloud. Therefore, to overcome these limitations, the following requirements should be met to design a LiDAR point cloud descriptor.

- Discriminative as well as compact: For fast save and retrieval of places, the summarization of a LiDAR point cloud should be compact such as a short length of a vector (e.g., 128- or 256-dimensional vector) or binary description. Discriminateness means every place can be well separated without confusion (This confusion is called perception aliasing). For SLAM, since false loop detection is critical for robot to estimate its position, this confusion should be avoid. However, because compactness and discriminative are in a trade-off relationship, we may lose some information with attempting to represent a point cloud with less space. Therefore, it is required to design a discriminative but sufficiently compact descriptor.
- Robust (invariant) to environmental condition changes: A point cloud descriptor should be not only discriminative but also invariant for measurements taken at the same place on different dates. This may not be required for online place recognition, but it is a heavily required qualification for robot robustness in outdoor.

### Taxonomy of LiDAR-based coarse localization methods

In this subsection, we provide a taxonomy of existing LiDAR-based coarse localization methods. The visualization of the taxonomy is given in Fig. 2.7.

First, the way of describing a point cloud can be categorized into two main streams; *local description* and *global description*. The word local and global means:

- Local Description: The point cloud is described by a set of local descriptors; a local descriptor is normally a feature vector extracted from a part (e.g., a single keypoint [8] or a segment [15, 16, 69]) of the scene.
- Global Description: The point cloud is represented by a single summarization. The single description summarizes the whole point cloud.

Due to the nature of the sparse point clouds obtained from 3D LiDAR in the outdoor urban environment, local descriptors are susceptible to noise and less discriminative. Therefore, in recent years, many global descriptors have been developed.

Since global descriptors can summarize a single place as a single descriptor, coarse localization has been performed by storing whole descriptors in a database for convenience and finding a candidate nearest (i.e., whose distance is minimum and underneath a certain threshold) to a query. We call this category *pairwise comparison with a databases*.

The first contribution of this thesis aims to develop a new point cloud descriptor and matching scheme belongs to this pairwise comparison family. Unlike conventional methods of this category that

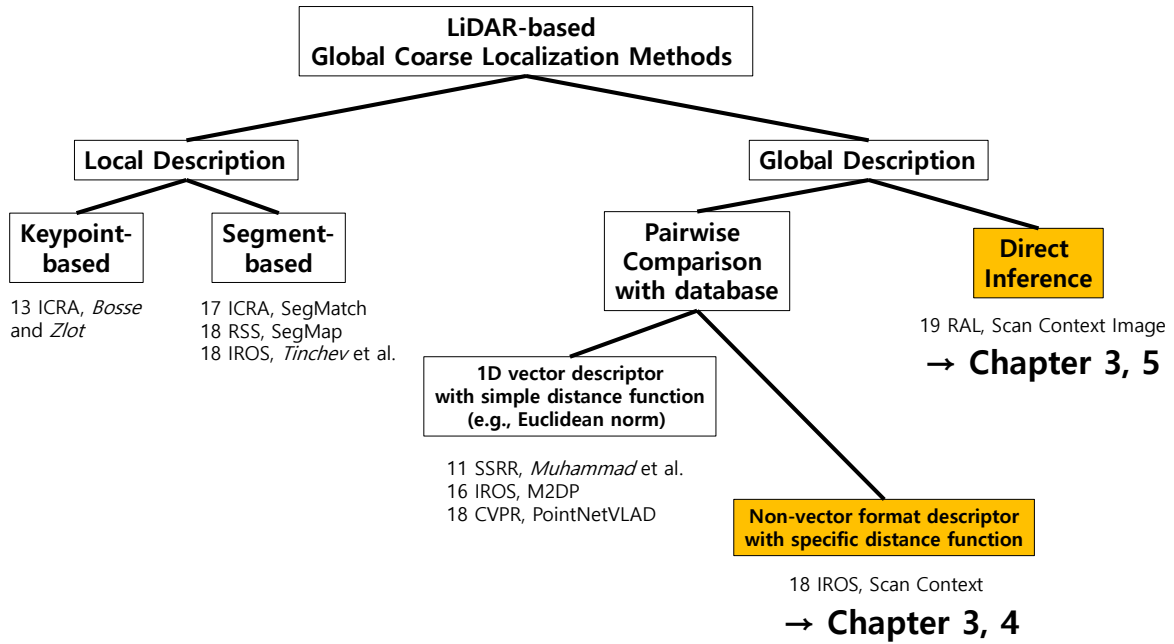


Figure 2.7: Taxonomy of robot localization methods using point cloud from LiDAR sensors.

concentrate on summarizing compactly into a 1D vector and use naive matching (e.g., using Euclidean norm), we develop a novel descriptor that contains more information, which is not a 1D vector form, and propose special matching algorithms to enhance both time efficiency and discriminative. The details are given in chapter §2.1.2.

The second contribution of this thesis belongs to a completely different category, *direct inference*. This category literally directly infers a current place of a robot. That is, it does not need to store descriptors in the database and not compare the query against the whole database. We make this possible by changing the formulation of the global coarse localization problem from place retrieval (before) to place classification (after). With the recent development of deep learning, it became possible to learn place information and save it in the network. We have knowledge (i.e., weights of the neural network), not a database. The details are given in chapter §2.1.3.

## Chapter 3. Scan Context: 3D Isovist for Robot Localization

### 3.1 Isovist

A term, *isovist*, was originally coined and developed from architecture [6]. This is a theory of visibility of an observer in a space (i.e., at a location within certain surrounding structures). In urban space design and landscape analysis researches, it is believed that visibility at a given space can affect the characteristics or even determine functions of that space. For example, in the middle of a wide square, an observer’s visibility will be high (or can be said the observer’s visible space is large). On the other hand, among the tall buildings, the observer would feel closed (that is, the observer’s visible space will be narrow or small).

In robotics this term, *isovist*, may be unfamiliar. However, a main goal of this thesis is to combine the concept of observer visibility, *isovist*, with a sensor measurements from LiDAR; thus we would like to say that ‘Robot localization meets 3D Isovist’ and the meaning of this sentence will be clear through this chapter. This resulted in a more efficient and effective point cloud descriptor than existing ones. Thus before we introduce our point cloud descriptor, we thought introduction of the concept of *isovist* first might be helpful in understanding our intuition.

In this section, we introduce the definition of *isovist* and then introduce our novel point cloud descriptor, *Scan Context*. Finally, we illuminate the meaning of the relationship between *isovist* and *Scan Context* in each domain (i.e., space analysis and robot localization).

#### 3.1.1 Introduction

##### Definition of Isovist

Benedikt [6] proposed a milestone study about *isovist*. *Isovist* is defined as a visible polygon within a given space. Thus this polygon varies with respect to the observer’s location as Fig. 3.1.

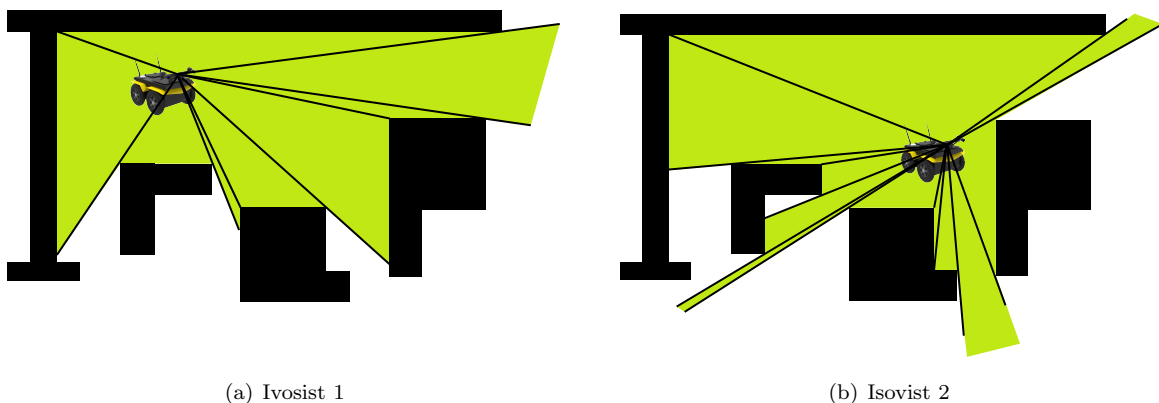


Figure 3.1: These two figures show the different cases of *isovist* polygon depending on the location of the robot. A green polygon in each figure is a *isovist*. In this figure the *isovist* is 2D, so called 2D *isovist*, since we assume that the robot exists in the 2D plane. As depicted, 2D *isovist* can mathematically be defined as a set of triangles.

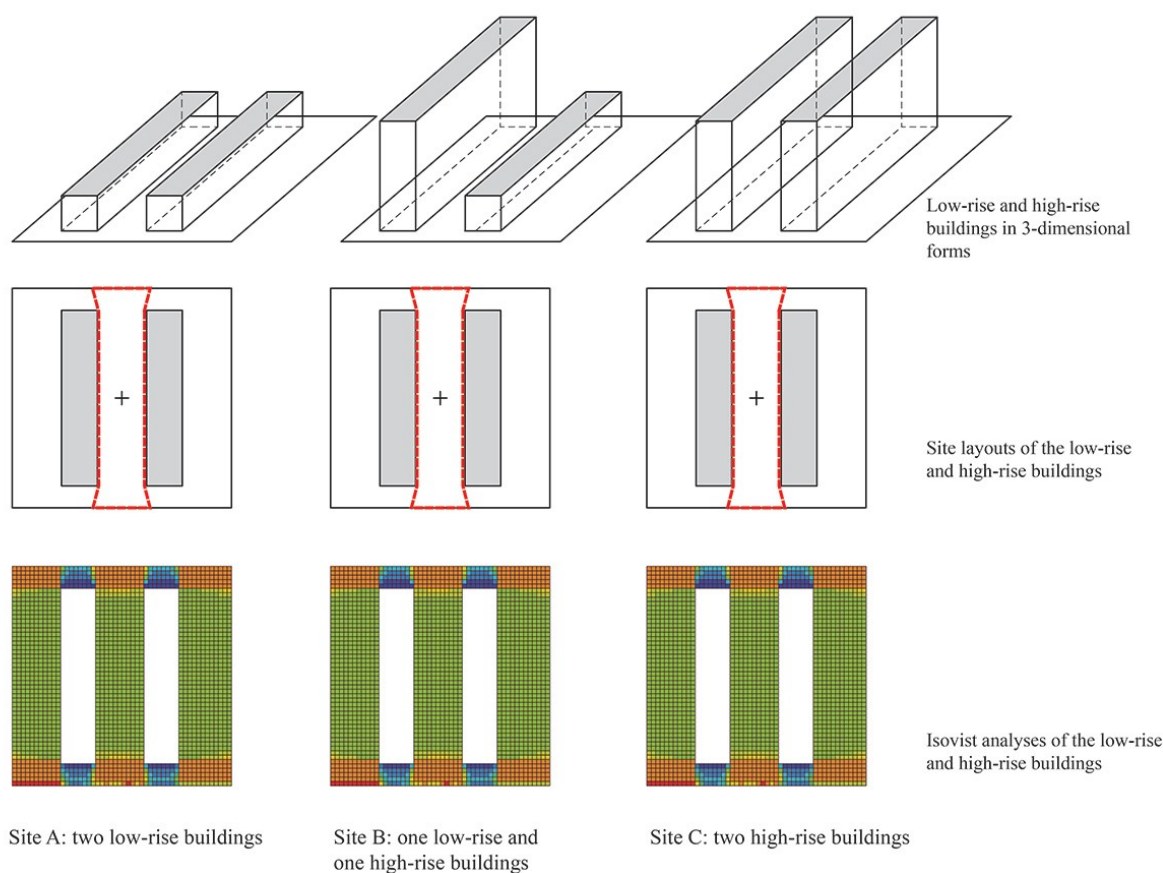


Figure 3.2: Limitation of 2D isovist. 2D isovist cannot captures a difference of 3D structures. Site A, B, and C have buildings of different heights, but their 2D isovist is the same.

Isovist have been used in urban design and space analysis since many useful measures can be derived from this visible polygon, such as area (or volume for 3D case) of the polygon, perimeter, convexity, or skewness [50].

### Limitation of 2D Isovist: Necessity of 3D Isovist

However, our real world is 3D, not 2D, and do not exists in a 2D plane. The last row (bottom) of Fig. 3.2 shows heatmaps of the area for each environment. 2D isovist is not able to distinguish the difference of 3D structures as Fig. 3.2.

Therefore, we need to define a 3D isovist that reflects the influence of 3D structures in 3D space. The underlying concept was early proposed in [6], but it took more time to actually implement it. Yang et al. [76] proposed a method to define 3D isovist based on Geographic Information System (GIS). They proposed model-based (using DEM) 3D isovist and showed empirical space analysis results by using volume as a measure of isovist.

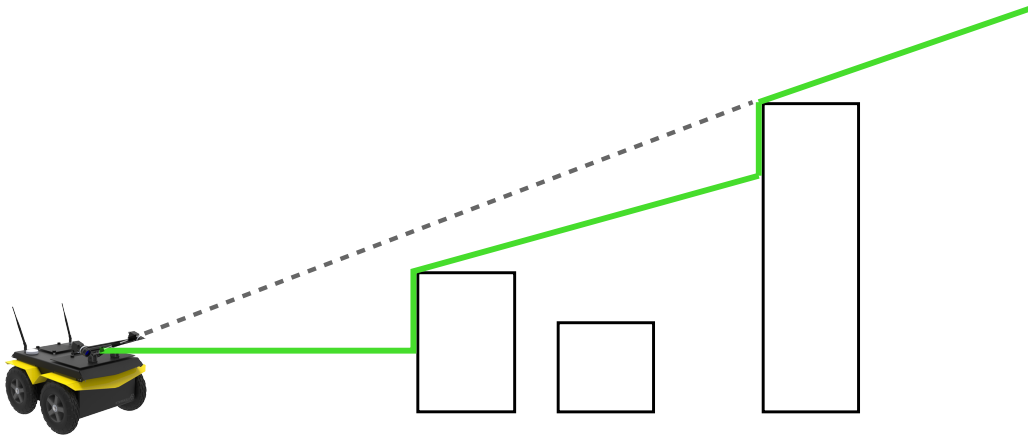


Figure 3.3: This figure shows an example of isovist polygon of a single line of sight and 3D isovist can be considered as an integration of this isovist of line of sight. Green polygon is the isovist.

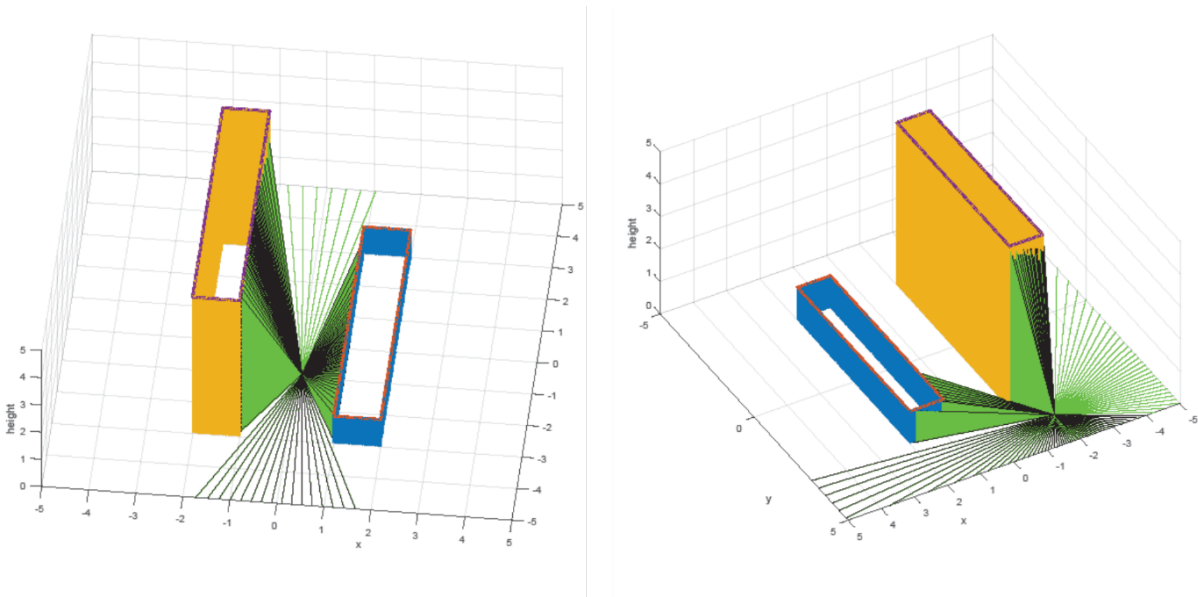


Figure 3.4: Example of 3D isovist on a DEM

### 3.1.2 Sensor data-driven 3D Isovist

#### Existing 3D Isovist: Model-based

The example of calculating the 3D isovist on a DEM as proposed in [50, 76] is shown in Fig. 3.4. Morello and Ratti defined an isovist of a single line of sight as Fig. 3.3 and represent a 3D isovist by integrating the isovist of each line of sight along omnidirectional sights.

Despite the aforementioned need, the reasons why researchers at urban design and landscape analysis struggle to study 3D isovist are due to these three reasons itemized as below. Because the existing 3D isovist is model-based, these reasons are considered limitations of model-based methods.

- Not real. GIS-based definition of isovist only reflects buildings assumed to be voxel-shaped. In the real world, however, non-building objects such as trees and signs also have an impact on an observer's visibility.



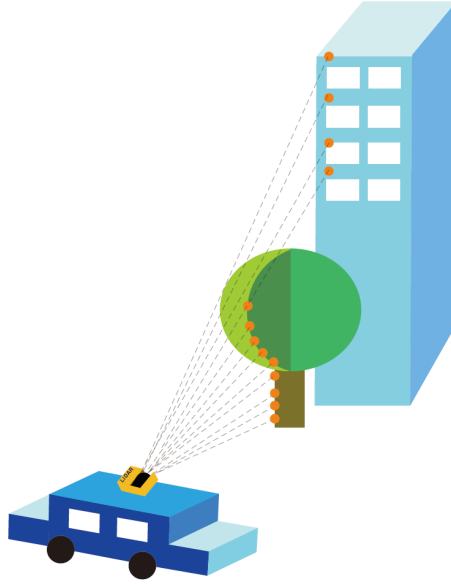


Figure 3.5: Visualization of a LiDAR scan. LiDAR sensor on a car can captures directly visible parts with any shape and without a prebuilt model. The points of orange color are a indeed reflected from the surface of the part of surrounding objects (i.e., a tree and a higher part of a building in this figure) and only sensed by the robot (car).

- Expensive. Prior to the analysis, building a city scale model should be preceded. DEM data can be obtained from aerial LiDAR, which is expensive to operate.
- Non-dynamic. From the above two reasons, model-based isovist can hardly reflect the dynamicity of the environment. As time and season change, the amount of tree bushes changes, and buildings could disappear or be newly made. However, since the cost of constructing the model is high, it is not easy to update the map frequently and consequently resulted in 3D isovist may be easily outdated.

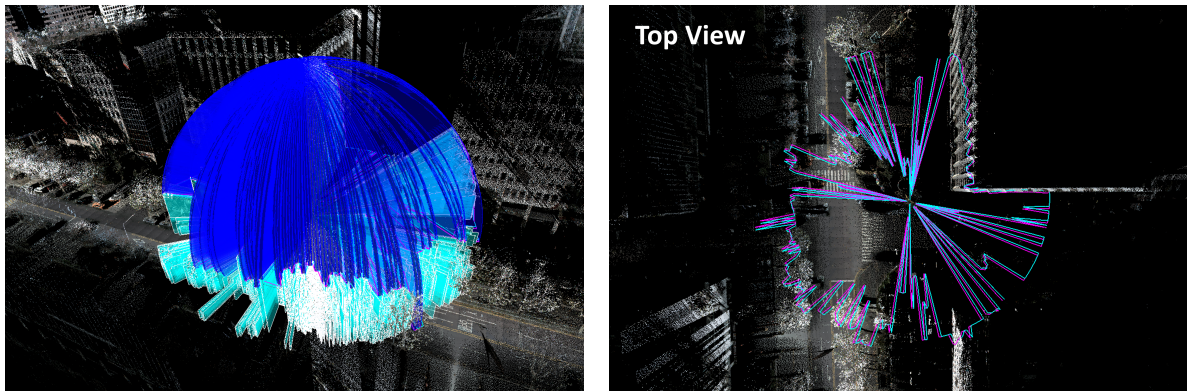
### Robot's visibility

On the other hand, a robot equipped with LiDAR sensors can overcome all of the aforementioned limitations of the model-based isovist for visibility analysis. The robot does not know the actual shape of the surrounding environment or buildings (e.g., tree and building in Fig. 3.5), but the temporal visible parts can be captured precisely (e.g., orange points in Fig. 3.5). This is all of the information needed for visibility analysis, and is more accurate than model-based isovist (i.e., green line in Fig. 3.3).

### Sensor data-driven 3D Isovist

Therefore, we now define isovist as a set of points actually observed in the line of sight from a 3D LiDAR sensor and call it *sensor data-driven 3D isovist*. 3D isovist is thus defined as a set of points for each line of sight.

Fig. 3.6 shows the sensor data-driven 3D isovist using a real-world point cloud data, which is captured from the area of Yeouido, Seoul, South Korea. Fig. 3.7 zooms in for a close-up of the 3D isovist in Fig. 3.6.



(a) Sensor data-driven 3D isovist using point cloud data captured from 3D LiDAR sensors

(b) Top view of the sensor data-driven 3D isovist

Figure 3.6: Example of sensor data-driven 3D isovist. In Fig. 3.6(b) the shape of the isovist is more accurate and realistic than model-based methods [50].

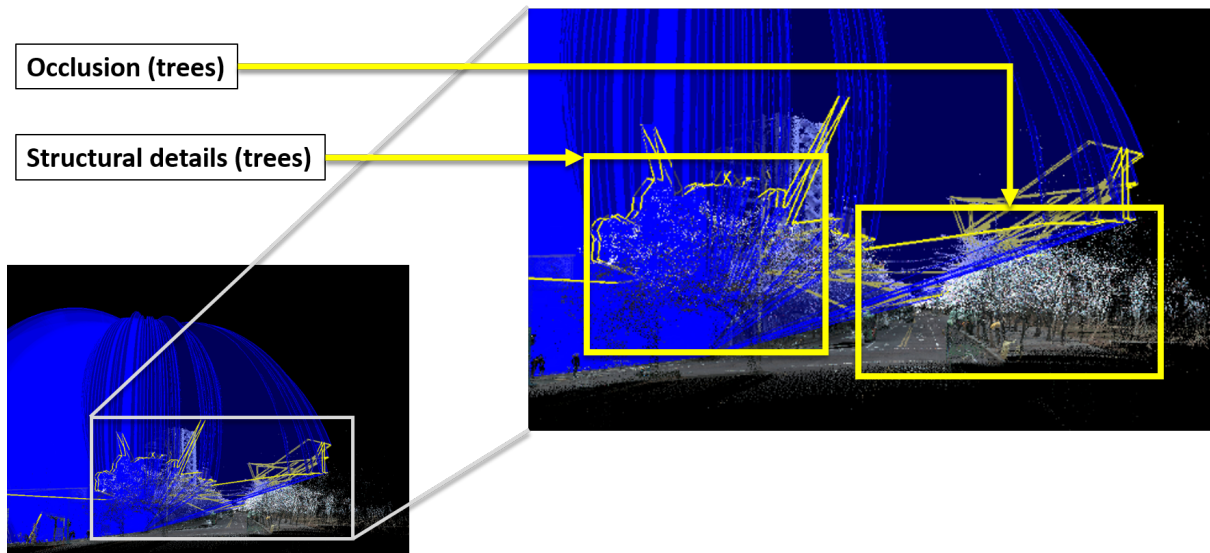


Figure 3.7: Advantages of sensor data-driven 3D isovist. We can find the realistic visible volume from the location is sometimes highly decreased by non-building objects (e.g., trees in this figure) and their view-limiting shape is unstructured and complex, which is hard for a model to predict.

3D isovist defined using a point cloud data from a LiDAR sensor can capture the surrounding structures as they are seen.

### Scan Context: Preview

Sensor data-driven 3D isovist is, however, complex than the model based 3D isovist with respect to the mathematical formulation since it has more accurate but complex shape so that hard to define its polygon. Therefore, to relieve the complexity, we propose a method to summarize the information of a visible polygon of a line of sight into a 1D vector to relieve instead of defining the geometric shape of a visible polygon

Our core idea is that ‘information of visible part is indeed important and we do not need an actual

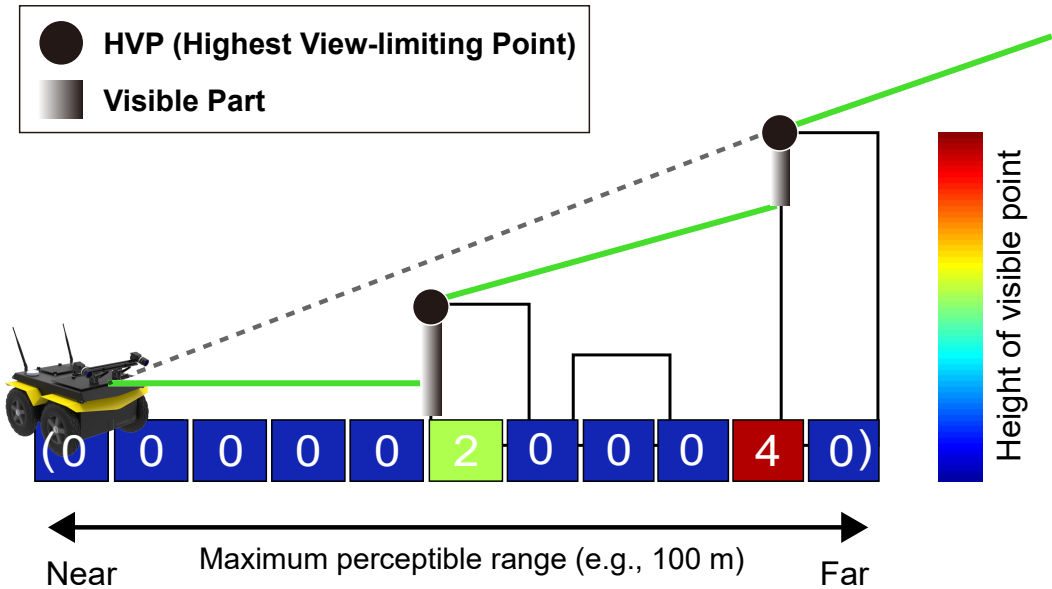


Figure 3.8: Preview of Scan Context. We encode a complex 3D geometric shape acquired from the actual LiDAR measurements into a 1D vector via leaving only visible part information; that is the maximum height of points within a discretized bin.

geometrical shape’. Fig. 3.8 impose the core idea. Instead of defining the shape of the visible polygon (green line), we discretize the sensing range of an a line of sight (a single element is called *bin*.) and encoded only the information corresponding to the visible part in each bin. We thought that the key information of a visible bin is the height of the highest point, which is called Highest View-limiting Point (HVP).

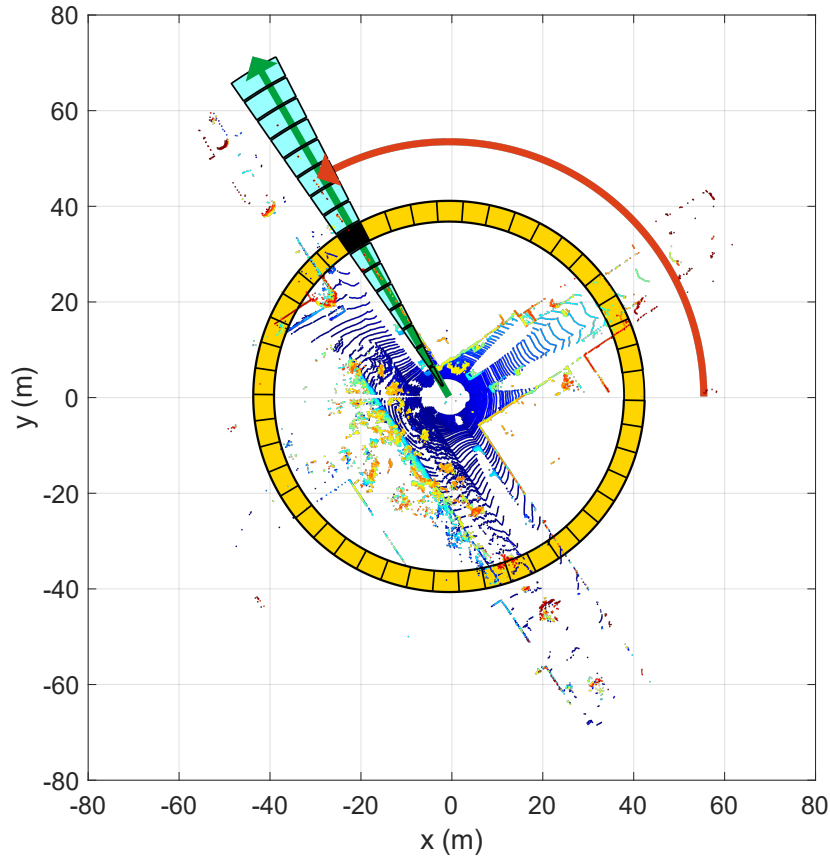
Thus, we are now able to represent the line of sight isovist as a 1D vector. sensor data-driven 3D isovist is easily defined because it is just an azimuthal integration version of Fig. 3.8. We will soon see in the next section that 3D isovist is our novel point cloud descriptor, Scan Context.

## 3.2 Scan Context (SC): Egocentric Place Descriptor

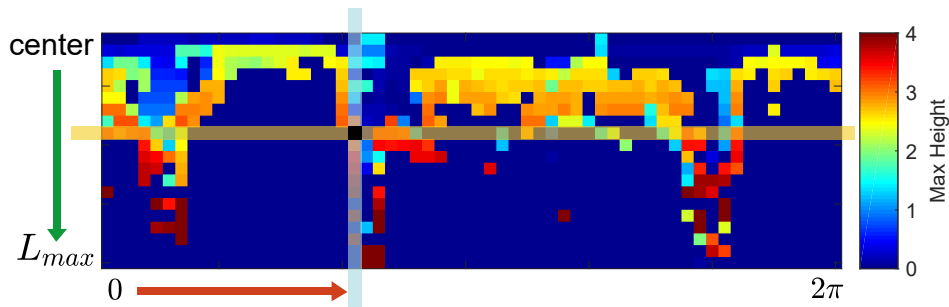
In this section, we introduce our main contribution, a novel point cloud descriptor called Scan Context. First, the definition of Scan Context is proposed. Then, a few characteristics of this novel representation are discussed.

### Definition of Scan Context

We define a place descriptor called *Scan Context* for outdoor place recognition. The key idea of a scan context is inspired by *Shape Context* [5] proposed by Belongie et al., which encodes the geometrical shape of the point cloud around a local keypoint into an image. While their method simply counts the number of points to summarize the distribution of points, ours differs from theirs in that we use a maximum height of points in each bin. The reason for using the height is to efficiently summarize the vertical shape of surrounding structures without requiring heavy computations to analyze the characteristics of the point cloud. In addition, the maximum height says which part of the surrounding structures is visible



(a) Bin division along azimuthal and radial directions



(b) Scan context

Figure 3.9: Two-step Scan Context creation. Using the top view of a point cloud from a 3D scan (a), we partition ground areas into bins, which are split according to both azimuthal (from 0 to  $2\pi$  within a LiDAR frame) and radial (from center to maximum sensing range) directions. We refer to the yellow area as a *ring*, the cyan area as a *sector*, and the black-filled area as a *bin*. Scan context is a matrix as in (b) that explicitly preserves the absolute geometrical structure of a point cloud. The ring and sector described in (a) are represented by the same-colored column and row, respectively, in (b). The representative value extracted from the points located in each bin is used as the corresponding pixel value of (b). We use the maximum height of points in a bin.

from the sensor. This egocentric visibility has been a well-known concept in the urban design literature for analyzing an identity of a place [6, 50].

Similar to shape context [5], we first divide a 3D scan into azimuthal and radial bins in the sensor coordinate, but in an equally spaced manner as shown in Fig. 3.9(a). The center of a scan acts as a global keypoint and thus we refer to a scan context as an egocentric place descriptor.  $N_s$  and  $N_r$  are the number of sectors and rings, respectively. That is, if we set the maximum sensing range of a LiDAR sensor as  $L_{max}$ , the radial gap between rings is  $\frac{L_{max}}{N_r}$  and the central angle of a sector is equal to  $\frac{2\pi}{N_s}$ .

Therefore, the first process of making a scan context is to partition whole points of a 3D scan into mutually exclusively separated point clouds as shown in Fig. 3.9(a).  $\mathcal{P}_{ij}$  is the set of points belonging to the bin where the  $i$ th ring and  $j$ th sector overlapped. The symbol  $[N_s]$  is equal to  $\{1, 2, \dots, N_s-1, N_s\}$ . Therefore, the partition is mathematically

$$\mathcal{P} = \bigcup_{i \in [N_r], j \in [N_s]} \mathcal{P}_{ij} . \quad (3.1)$$

Because the point cloud is divided at regular intervals, a bin far from a sensor has a physically wider area than a near bin. However, both are equally encoded into a single pixel of a scan context. Thus, a scan context compensates for the insufficient amount of information caused by the sparsity of far points and treats nearby dynamic objects as sparse noise.

After the point cloud partitioning, a single real value is assigned to each bin by using the point cloud in that bin:

$$\phi: \mathcal{P}_{ij} \rightarrow \mathbb{R} , \quad (3.2)$$

and we use a maximum height, which is inspired from the urban visibility analysis [6, 50]. Thus, the bin encoding function is

$$\phi(\mathcal{P}_{ij}) = \max_{\mathbf{p} \in \mathcal{P}_{ij}} z(\mathbf{p}) , \quad (3.3)$$

where  $z(\cdot)$  is the function that returns a z-coordinate value of a point  $\mathbf{p}$ . We assign a zero for empty bins. For example, as seen in Fig. 3.9(b), a blue pixel in the scan context means that the space corresponding to its bin is either free or not observed due to occlusions.

From the foregoing processes, a scan context  $I$  is finally represented as a  $N_r \times N_s$  matrix as

$$I = (a_{ij}) \in \mathbb{R}^{N_r \times N_s} , \quad a_{ij} = \phi(\mathcal{P}_{ij}) . \quad (3.4)$$

## Resolution of Scan Context

There are three user parameters for Scan Context; the number of sectors ( $N_s$ ), the number of rings ( $N_r$ ), and the maximum sensing range ( $L_{max}$ ). The more subdivided the space (the more finer resolution), the more precise it is possible to express. Abstraction power and detail preservation are in trade-off. However, it is not always good to summarize the space in detail. It is likely to be vulnerable to noise when too fine.

Fig. 3.10(b) displays several Scan Contexts of various resolutions, which start from  $(N_s, N_r) = (15, 5)$  and become fine by 2 times. Here we fixed  $L_{max}$  to 80 m. We use resolutions (60, 20) and (120, 40) for application 1 (chapter §4) and application 2 (chapter §5), respectively.

## Viewpoint Information in Scan Context

A common global descriptor for a point cloud is designed to be viewpoint invariant to allow robots to recognize the same location in any direction. However, the point cloud is summarized in viewpoint invariant, but the viewpoint information itself is lost because existing methods usually use statistical

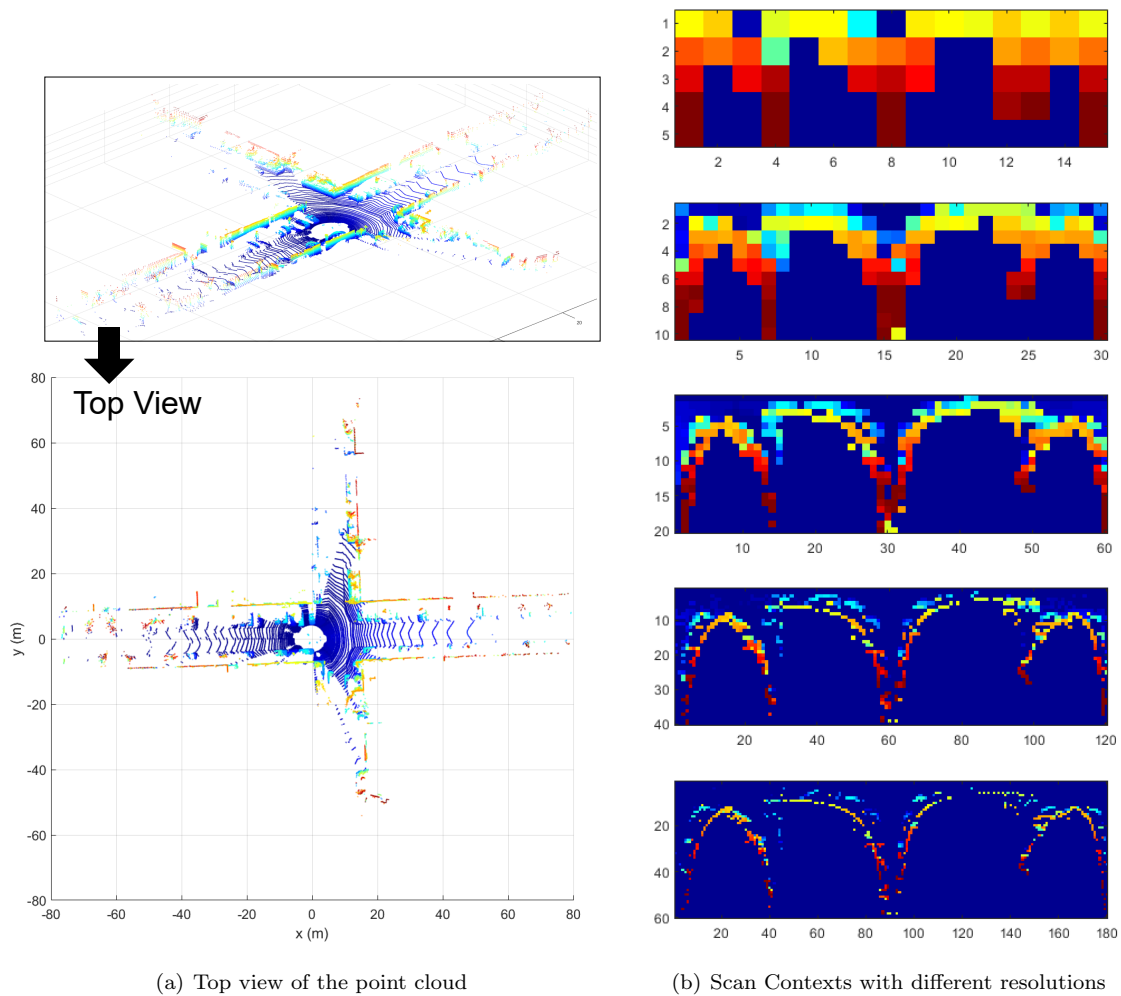


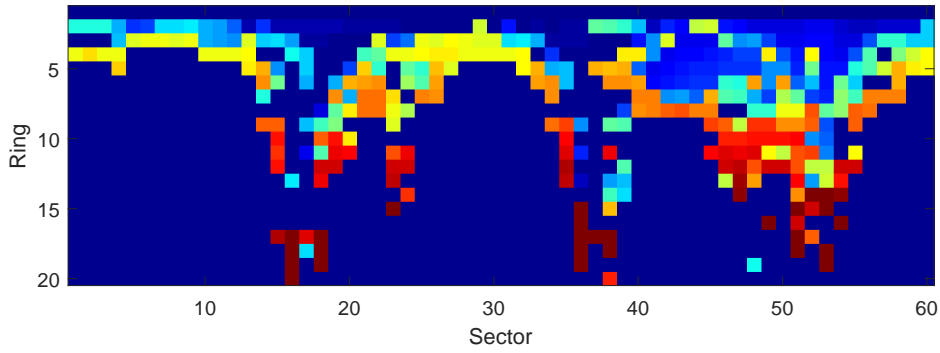
Figure 3.10: An example of Scan Context

techniques such as histogram [51, 74]. The details of existing point cloud descriptor are introduced in section §4.2.

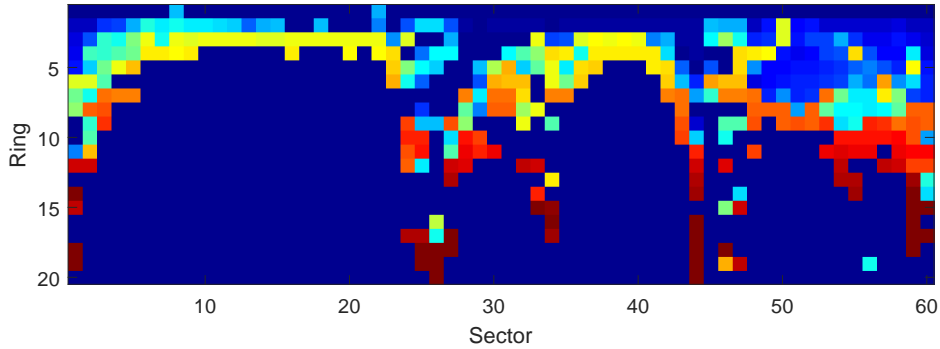
But the Scan Context starts with the opposite question: preserving the original viewpoint information. The column order of a Scan Context is the viewpoint. The column of the candidate scan context may be shifted even in the same place, since a viewpoint of a LiDAR changes for different places (e.g., revisit in an opposite direction or rotation at a corner). Fig. 3.11 illustrates such cases. Since a scan context is the representation dependent on the sensor location, the row order is always consistent. However, the column order could be different if the LiDAR sensor coordinate with respect to the global coordinate changed.

The two advantages of preserving viewpoint information are:

1. Robust viewpoint invariant place recognition. Ours shows the improved performance over existing descriptors, in §4, especially for reverse detection.
2. Preservation of local geometry. The local geometry corresponds to a local patch in a Scan Context (Scan Context is 2D matrix so we can consider it as an image). This consistency of local geometry is suitable for CNN-based networks to learn patterns. Existing descriptors lose the local geometry entirely since they summarize information statistically or consume points directly to learn a pattern



(a) The query scan context (3280<sup>th</sup> scan, KITTI00)



(b) The detected scan context (2345<sup>th</sup> scan, KITTI00)

Figure 3.11: Example of scan contexts from the same place with time interval. The change of the sensor viewpoint at the revisit causes column shifts of the scan context as in (a). However, the two matrices contain similar shapes and show the same row order.

of the geometry of an entire point cloud in an end-to-end manner, which is not suitable for place learning because a robot may not visit the same place a few hundred or thousand times for learning.

### Scan Context and 3D Isovist: Summary

We have examined how a novel point cloud descriptor, Scan Context, have emerged from the following question: How can we effectively represent the sensor data-driven 3D isovist?

We will show in §4 and §5 that this descriptor effectively summarizes a place and shows good performance in robot localization for both online place recognition and long-term localization.

That is, 3D isovist was originally a concept defined in urban analysis, but we can say that now 3D isovist can be used for robot localization. Therefore, we want to conclude this chapter by summarizing the relationship between Scan Context and 3D isovist as follows.

- For space analysis: Scan Context is data-driven 3D Isovist. Scan Context is the effective and efficient representation of 3D Isovist polygon.
- For mobile robotics: Scan Context is a visibility-based place fingerprint.

## Chapter 4. Application 1: Online Place Recognition

Compared to diverse feature detectors and descriptors used for visual scenes, describing a place using structural information is relatively less reported. Recent advances in SLAM provides dense 3D maps of the environment and the localization is proposed by diverse sensors. Toward the global localization based on the structural information, we propose *Scan Context*, a non-histogram-based global descriptor from 3D LiDAR scans. Unlike previously reported methods, the proposed approach directly records a 3D structure of a visible space from a sensor and does not rely on a histogram or on prior training. In addition, this approach proposes the use of a similarity score to calculate the distance between two scan contexts and also a two-phase search algorithm to efficiently detect a loop. Scan context and its search algorithm make loop-detection invariant to LiDAR viewpoint changes so that loops can be detected in places such as reverse revisit and corner. Scan context performance has been evaluated via various benchmark datasets of 3D LiDAR scans, and the proposed method shows a sufficiently improved performance.

### 4.1 Introduction

In many robotics applications, place recognition is the important problem. For SLAM, in particular, this recognition provides candidates for loop-closure, which is essential for correcting drift error and building a globally consistent map [9]. While the loop-closure is critical for robot navigation, wrong registration can be catastrophic and careful registration is required. Visual recognition is popular together with the widespread use of camera sensors, however, it is inherently difficult due to illumination variance and short-term (e.g., moving objects) or long-term (e.g., seasons) changes. Similar environments may occur at different locations often causing perception aliasing. Therefore, recent literature has focused on robust place recognition by examining representation [25] and resilient back-end [42].

Unlike these visual sensors, LiDARs have recently garnered attention due to their strong invariance to perceptual variance. In the early days, conventional local keypoint descriptors [5, 32, 58, 59], which were originally designed for the 3D model in computer vision, have been used for place recognition in spite of their vulnerability to noise. LiDAR-based methods for place recognition have been widely proposed in robotics literature [27, 29, 66]. These works focus on developing descriptors from structural information (e.g., point clouds) in both local [66] and global manners [27].

There are two issues that the existing LiDAR-based place recognition methods have been trying to overcome. First, the descriptor is required to achieve rotational invariance regardless of the viewpoint changes. Second, noise handling is the another topic for these spatial descriptors because the resolution of a point cloud varies with distance and normals are noisy. The existing methods mainly use the histogram [29, 51, 74] to address the two aforementioned issues. However, since the histogram method only provides a stochastic index of the scene, describing the detailed structure of the scene is not straightforward. This limitation makes the descriptor less discernible for place recognition problem, causing potential false positives.

In this chapter we present *Scan Context*, a novel spatial descriptor with a matching algorithm, specifically targeting outdoor place recognition using a single 3D scan. Our representation encodes a whole point cloud in a 3D scan into a matrix (Fig. 3.9). The proposed representation describes egocentric



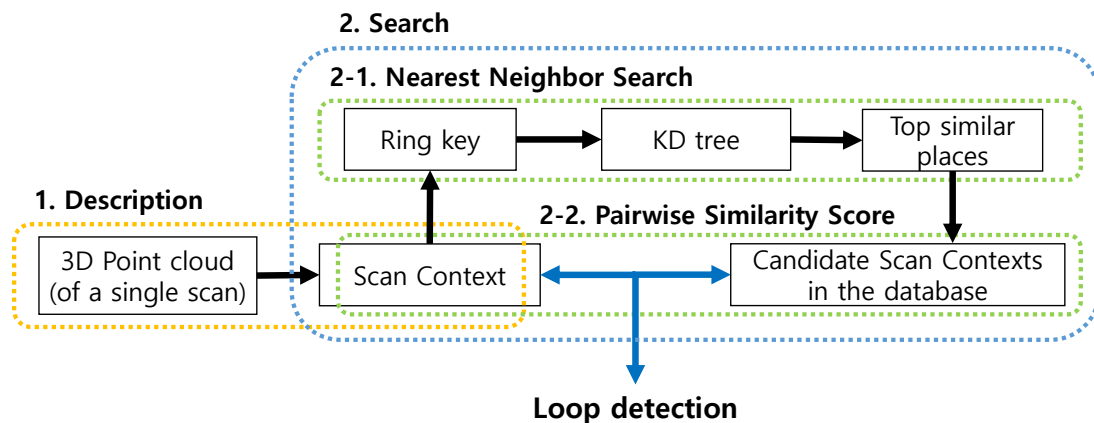


Figure 4.1: Algorithm overview. First, a point cloud in a single 3D scan is encoded into scan context. Then,  $N_r$  (the number of rings) dimensional vector is encoded from the scan context and is used for retrieving the nearest candidates as well as the construction of the KD tree. Finally, the retrieved candidates are compared to the query scan context. The candidate that satisfies the acceptance threshold and is closest to the query is considered the loop.

2.5D information. Contribution points of the proposed method are:

- *Efficient bin encoding function.* Unlike existing point cloud descriptors [5, 27], the proposed method needs not count the number of points in a bin, instead it proposes a more efficient bin encoding function for place recognition. This encoding presents invariance to density and normals of a point cloud.
- *Preservation of internal structure of a point cloud.* As shown in Fig. 3.9, each element value of a matrix is determined by only the point cloud belonging to the bin. Thus, unlike [29], which depicts the relative geometry of points as a histogram and loses points' absolute location information, our method preserves the absolute internal structure of a point cloud by intentionally avoiding using a histogram. This improves the discriminative capability and also enables viewpoint alignment of a query scan to a candidate scan (in our experiments,  $6^\circ$  azimuth resolution) while a distance is calculated. Therefore, detecting a reverse direction loop is also possible by using scan context.
- *Effective two-phase matching algorithm.* To achieve a feasible search time, we provide a rotational invariant subdescriptor for first nearest neighbor search and combine it with pairwise similarity scoring hierarchically, thus avoid searching all databases for loop-detection.
- *Thorough validation against other state-of-the-art spatial descriptors.* In the comparison to other existing global point cloud descriptors, such as M2DP [66], Ensemble of Shape Functions (ESF) [74], and Z-projection [51], the proposed approach presents a substantial improvement.

## 4.2 Related Work

Place recognition methods for mobile robots can be categorized into vision-based and LiDAR-based methods. Visual methods have been commonly used for place recognition in SLAM literatures [2, 14, 22]. FAB-MAP [14] increased robustness with the probabilistic approach by learning a generative model for the bag of visual words. However, visual representation has limitations such as vulnerability to light

condition change [71]. Several methods have been proposed to overcome these issues. SeqSLAM [48] proposed the route-based approach and showed far improved performance than FAB-MAP. SRAL [25] fused several different representation such as color, GIST [68], and HOG [53] for long-term visual place recognition.

LiDAR presents strong robustness to these perceptual changes described above. LiDAR-based methods are further categorized into local and global descriptors. Local descriptors, such as PFH [58], SHOT [59], shape context [5], or spin image [32], first find a keypoint, separate nearby points into bins, and encode a pattern of surrounding bins into a histogram. Steder et al. proposed the place recognition method [66] using point features and the gestalt descriptor [7] in bag of words manner.

These keypoint descriptors, however, revealed limitations since they were originally devised for 3D model part matching not for place recognition. For example, the density of a point cloud in a 3D scan (e.g., from VLP-16) varies with respect to the distance from a sensor, unlike the 3D model. Furthermore, normals of points are noisier than the model due to unstructured objects (e.g., trees) in the real world. Hence, local methods usually require normals of keypoints and thus are less suitable for place recognition in outdoor.

Global descriptors do not include the keypoint detecting phase. GLARE [29] and its variations [34, 57] encoded the geometric relationship between points into a histogram in lieu of searching for the keypoint and extracting the descriptor. ESF [74] used concatenation of histograms made from shape functions. Muhammad and Lacroix proposed Z-projection [51], which is a histogram of normal vectors, and a double threshold scheme with two distance functions. He et al. proposed M2DP [27], which projects a whole 3D point cloud of a scan to multiple 2D planes and extracts a 192 dimensional compact global representation. M2DP showed higher performance than the existing point cloud descriptors and robustness against noise and resolution changes. As introduced in this paragraph, global descriptors have typically used histograms. Recently, SegMatch [15] introduced a segment-based matching algorithm. This is a high-level perception but requires a training step, and points are needed to be represented in a global reference frame.

Meanwhile, Kim et al. proposed a map representation method [38] that captures both elevation and occupancy information within a grid map for robust localization.

In this chapter, we propose a novel place descriptor called *Scan Context* that encodes a point cloud of a 3D scan into a matrix. The scan context can be considered as an extension of the *Shape Context* [5] for place recognition targeting 3D LiDAR scan data. In detail, scan context has three components: the representation that preserves absolute location information of a point cloud in each bin, efficient bin encoding function, and two-step search algorithm.

### 4.3 Scan Context for Place Recognition

In this section, we describe scan context creation given a point cloud from a 3D scan and propose a measure that calculates the distance between two scan contexts. Next, the two-step search process is introduced. The overall pipeline of place recognition using scan context is depicted in Fig. 4.1. The *Scan Context* creation and validation can also be found in a video (YouTube link, [https://youtu.be/\\_etNafgQXoY](https://youtu.be/_etNafgQXoY))

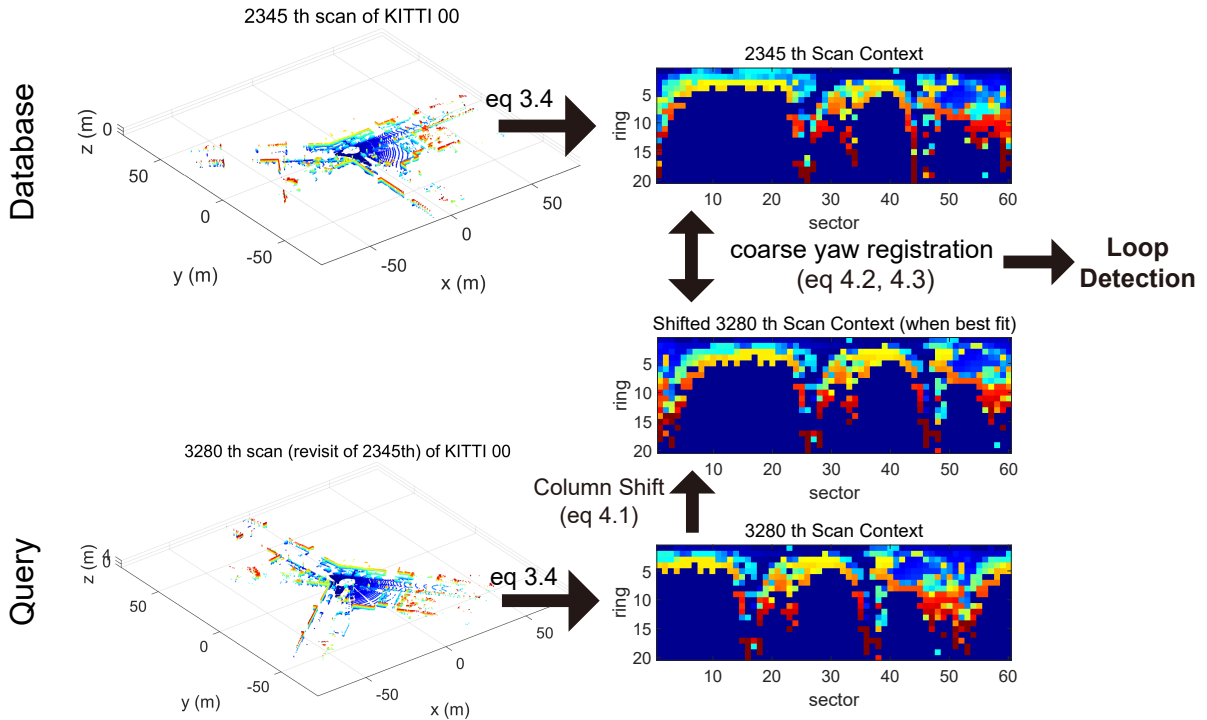


Figure 4.2: Example captures from Complex Urban Dataset [31]. With the advancement of SLAM technology, it is now easier to get a large scale point cloud map like this, which allows non-mapper robots to focus on localization.

#### 4.3.1 Similarity Score between Scan Contexts

Given a scan context pair, we then need a distance measure for the similarity of two places.  $I^q$  and  $I^c$  are scan contexts acquired from a query point cloud and a candidate point cloud, respectively. They are compared in a columnwise manner. That is, the distance is the sum of distances between columns at a same index. A cosine distance is used to compute a distance between two column vectors at the same index,  $c_j^q$  and  $c_j^c$ . In addition, we divide the summation by the number of columns  $N_s$  for normalization. Therefore, the distance function is

$$d(I^q, I^c) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left( 1 - \frac{c_j^q \cdot c_j^c}{\|c_j^q\| \|c_j^c\|} \right). \quad (4.1)$$

The column-wise comparison is particularly effective for dynamic objects by considering the consensus of throughout sectors. However, the column of the candidate scan context may be shifted even in the same place, since a viewpoint of a LiDAR changes for different places (e.g., revisit in an opposite direction or corner). Fig. 3.11 illustrates such cases. Since a scan context is the representation dependent on the sensor location, the row order is always consistent. However, the column order could be different if the LiDAR sensor coordinate with respect to the global coordinate changed.

To alleviate this problem, we calculate distances with all possible column-shifted scan contexts and find the minimum distance.  $I_n^c$  is a scan context whose  $n$  columns are shifted from the original one,  $I^c$ . This is the same task as roughly aligning two point clouds for yaw rotation at  $\frac{2\pi}{N_s}$  resolution. Then we

decide that the number of column shift for the best alignment (4.3) and the distance (4.2) at that time:

$$\begin{aligned} D(I^q, I^c) &= \min_{n \in [N_s]} d(I^q, I_n^c) , \\ n^* &= \operatorname{argmin}_{n \in [N_s]} d(I^q, I_n^c) . \end{aligned} \quad (4.2)$$

Note that this additional shift information may serve as a good initial value for further localization refinement such as Iterated Closest Point (ICP), as shown in Section 4.4.3.

For robust recognition over translation, we leverage scan context augmentation through *root shifting*. By doing so, acquiring various scan contexts from the raw scan under a slight motion perturbation becomes feasible. A single scan context may be sensitive to the center location of a scan under translational motion during revisit. For example, the row order of a scan context may not be preserved when revisiting the same place in a different lane. To overcome this situation, we translate a raw point cloud into  $N_{trans}$  neighbors ( $N_{trans} = 8$  used in the paper) depending on the lane level interval and store scan contexts obtained from these root-shifted point clouds together. We assumed that a similar point cloud is obtained even at the actual moved location, which is valid except for a few cases such as an intersection access point where a new space suddenly appears.

### 4.3.2 Two-phase Search Algorithm

Three main streams are typical when searching in the context of place recognition: pairwise similarity scoring, nearest neighbor search, and sparse optimization [79]. Our search algorithm fuses both pairwise scoring and nearest search hierarchically to achieve a reasonable searching time.

Since our distance calculation in (4.2) is heavier than other global descriptors such as [27, 51], we provide a two-phase hierarchical search algorithm via introducing ring key. Ring key is a rotation-invariant descriptor, which is extracted from a scan context. Each row of a scan context,  $r$ , is encoded into a single real value via ring encoding function  $\psi$ . The first element of the vector  $\mathbf{k}$  is from the nearest circle from a sensor, and following elements are from the next rings in order as illustrated in Fig. 4.3. Therefore, the ring key becomes a  $N_r$ -dimensional vector as (4.4):

$$\mathbf{k} = (\psi(r_1), \dots, \psi(r_{N_r})), \text{ where } \psi: r_i \rightarrow \mathbb{R} . \quad (4.4)$$

The ring encoding function  $\psi$  we use is the occupancy ratio of a ring using  $L_0$  norm:

$$\psi(r_i) = \frac{\|r_i\|_0}{N_s} . \quad (4.5)$$

Since the occupancy ratio is independent of the viewpoint, the ring key achieves rotation invariance.

Although being less informative than scan context, ring key enables fast search for finding possible candidates for loop. The vector  $\mathbf{k}$  is used as a key to construct a KD tree. At the same time, the ring key of the query is used to find similar keys and their corresponding scan indexes. The number of top similar keys that will be retrieved is determined by a user. These constant number of candidates' scan contexts are compared against the query scan context by using distance (4.2). The closest candidate to the query satisfying an acceptance threshold is selected as the revisited place:

$$c^* = \operatorname{argmin}_{c_k \in \mathcal{C}} D(I^q, I^{c_k}), \text{ s.t. } D < \tau , \quad (4.6)$$

where  $\mathcal{C}$  is a set of indexes of candidates extracted from KD tree and  $\tau$  is a given acceptance threshold.  $c^*$  is the index of the place determined to be a loop.

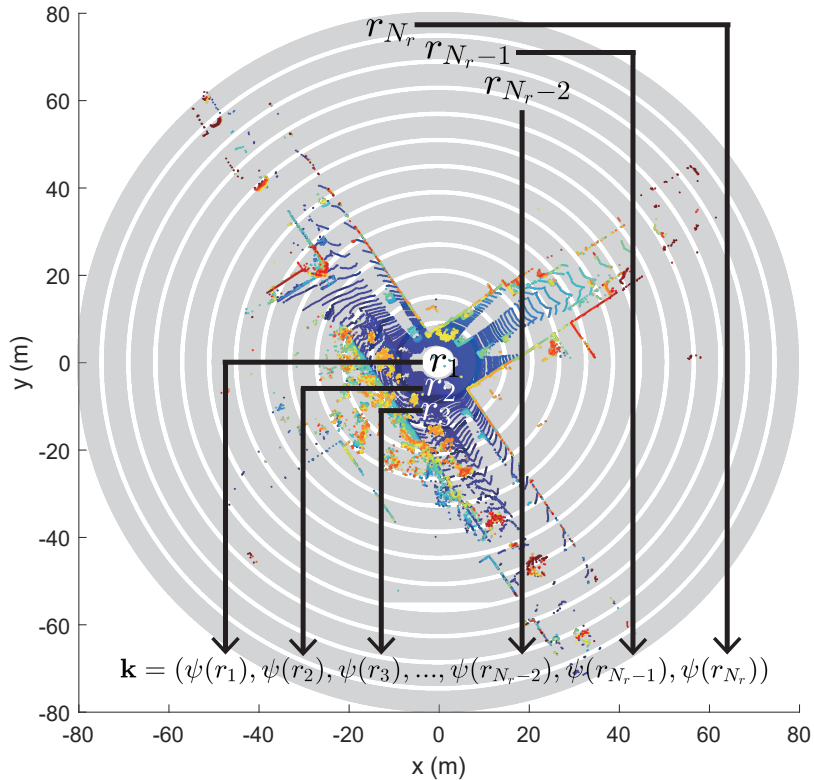


Figure 4.3: The ring key generation for the fast search.

## 4.4 Experimental Evaluation

In this section, our representation and algorithm are evaluated over various datasets and against other state-of-the-art algorithms. Since scan context is the global descriptor, the performance of our representation is compared to three other global representations using a 3D point cloud: M2DP [27], Z-projection [51], and ESF [74]. We use ESF in the Point Cloud Library (PCL) implemented in C++, Matlab codes of M2DP on the web<sup>1</sup> from the authors He et al., and implement Z-projection on Matlab ourselves. All experiments are carried out on the same system with an Intel i7-6700 CPU at 3.40GHz and 16GB memory.

### 4.4.1 Dataset and Experimental Settings

We use the KITTI dataset<sup>2</sup> [24], the NCLT dataset<sup>3</sup> [10], and the Complex Urban LiDAR dataset<sup>4</sup> [31] for the validation of our method. These three datasets are selected considering diversity, such as the type of the 3D LiDAR sensor (e.g., the number of rays, sensor mount types such as surround and tilted) and the type of loops (e.g., occurred at the same direction or the opposite direction called reverse loop). Characteristics of each dataset are summarized in Table 4.1. The term node means a single sampled place.

<sup>1</sup><https://github.com/LiHeUA/M2DP>

<sup>2</sup>[http://www.cvlibs.net/datasets/kitti/eval\\_odometry.php](http://www.cvlibs.net/datasets/kitti/eval_odometry.php)

<sup>3</sup><http://robots.engin.umich.edu/nclt/>

<sup>4</sup><http://irap.kaist.ac.kr/dataset/>




	KITTI (13 IJRR)	NCLT (16 IJRR)	ComplexUrban (18 ICRA)
LiDAR Type	64 ray	32 ray	Two tilted 16 ray
Environment Type	Suburban	Campus	Metropolitan <i>Difficult</i>
			

Figure 4.4: Selected datasets for the evaluation

### KITTI dataset

Among the 11 sequences having the ground truth of pose (from 00 to 10), the top four sequences whose the number of loop occurrences is highest are selected: 00, 02, 05, and 08. The sequence 08 has only reverse loops, and others have loop events with the same direction. The scans of the KITTI dataset had been obtained from the 64-ray LiDAR (Velodyne HDL-64E) located in the center of the car. Since the KITTI dataset provides scans with indexes, we use each bin file as a node directly.

### NCLT dataset

The NCLT dataset provides long-term measurements of different days along similar routes. Scans of the NCLT dataset were obtained from the 32-ray LiDAR (Velodyne HDL-32E) attached to a segway mobile platform. Four sequences are selected considering the number of loop occurrences and seasonal diversity. In this experiment, the scans are sampled at equidistant (2 m) intervals, and only those sampled scans are used as nodes for convenience.

### Complex Urban LiDAR dataset

The Complex Urban LiDAR dataset includes various complex urban environments from residential to metropolitan areas. Four sequences are selected considering the complexity and wide road rate provided by [31]. Among three sub-routes in the sequence 04, 04.0 and 04.1 are used in this experiment. The scans are sampled at 3 m intervals for convenience. The interesting fact is that this dataset uses two tilted LiDARs (Velodyne VLP-16 PUCK) for urban mapping. Thus, a single scan of this dataset is able to measure higher parts of structures but does not have a 360° surround view. To include more information in all directions, we merge the point clouds from both left and right tilted LiDARs and use them as a single scan to create a scan context.

Table 4.1: Selected dataset lists used in validation

Sequence Index	KITTI				NCLT				Complex Urban LiDAR			
	00	02	05	08	20120526	20120820	20120928	20130405	00	01	02	04
Total Length (m)	3714	4268	2223	3225	6345	6018	5579	4530	12020	11830	3020	6542
# of Nodes	4541	4661	2761	4071	3164	3001	2781	2259	3630	3266	862	2140
# of True Loops	790	309	493	332	810	526	635	275	361	383	125	150
Route Dir. on revisit	Same	Same	Same	Reverse	Both	Both	Both	Both	Same	Same	Both	Same

If a ground truth pose distance between the query and the matched node is less than 4 m, the detection is considered as true positive. In total 50 previously adjacent nodes are excluded from the search. The experiments for scan context are conducted with 10 candidates and 50 candidates from the KD tree, thus each method is called `scan context-10` and `scan context-50`, respectively. Unlike the scan context, which only compares with a constant number of candidates extracted from the KD tree, other methods (M2DP, ESF, and Z-projection) compare the query description to all in the database. In this chapter, we set parameters of scan context as  $N_s = 60$ ,  $N_r = 20$ , and  $L_{max} = 80$  m. That is, each sector has a  $6^\circ$  resolution and each ring has a 4 m gap. The number of bins of Z-projection is set as 100. We use the default parameters of the available codes for M2DP and ESF. For the computation efficiency, we downsample point cloud with  $0.6 \text{ m}^3$  grid for both scan context and M2DP, since He et al. [27] reported M2DP is robust to downsampling, whereas Z-projection and ESF use an original point cloud without downsampling because they are vulnerable to low density. We change only an acceptance threshold in the experiments.

#### 4.4.2 Precision Recall Evaluation

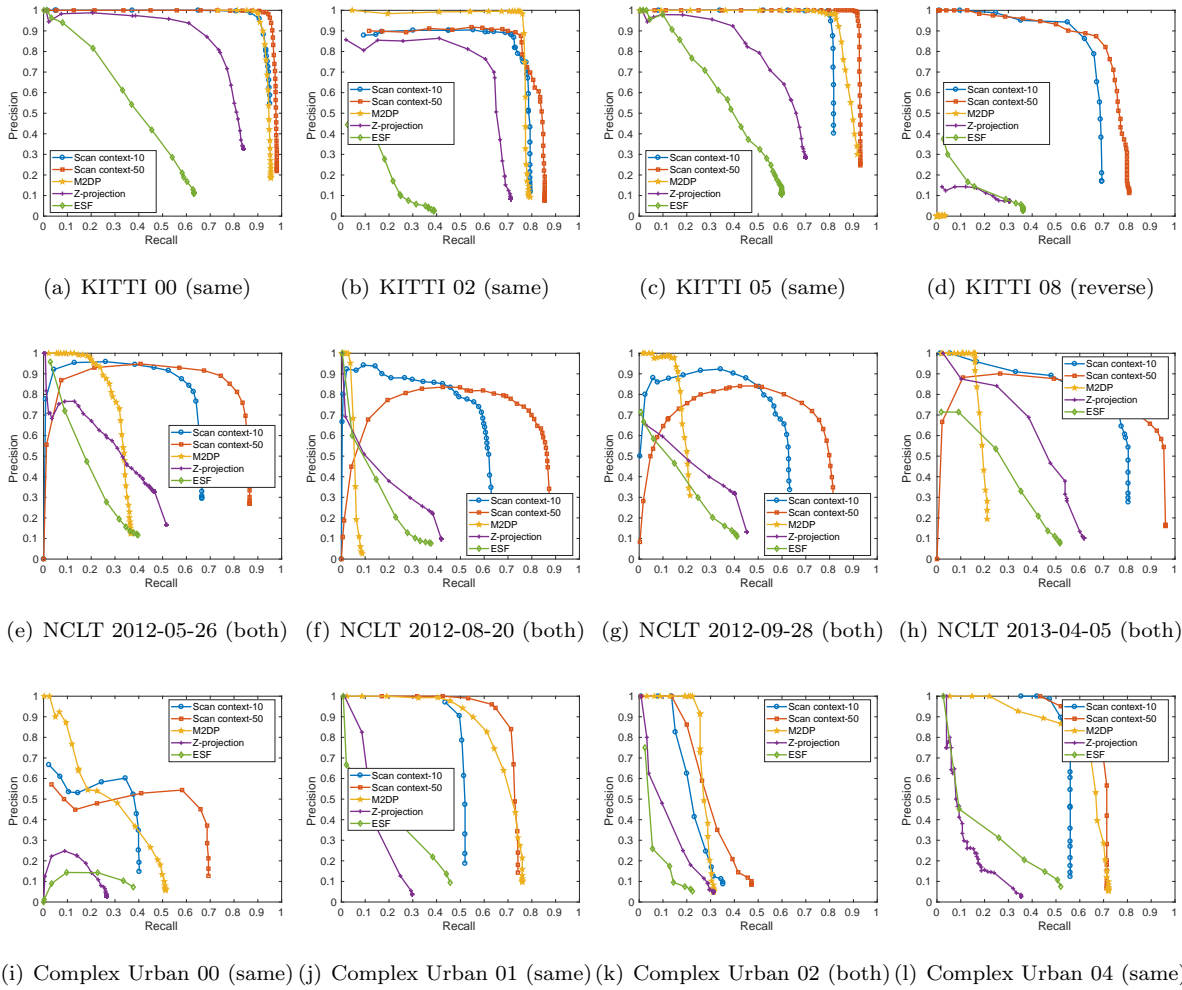


Figure 4.5: Precision-recall curves for the evaluation datasets. The route direction during the revisit is shown in parentheses.

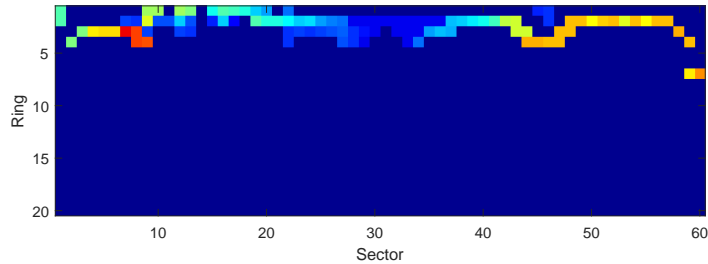


Figure 4.6: A challenging example captured from Complex Urban LiDAR dataset sequence 02. The road is so narrow in all directions that the amount of available information is too small.

The performance of *Scan Context* is analyzed using the precision-recall curve as in Fig. 4.5. Precision and recall are defined as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (4.7)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (4.8)$$

The histogram-based approaches, ESF and Z-projection, reported poor performances on all datasets. These methods rely on the histogram and distinguish places only when the structure of the visible space is substantially different. Unlike these histogram based methods, ours presented the meaningful performance for the entire data sequences. Overall, `scan context-50` always reveals better performance than `scan context-10`. The performance of scan context depends on the number of candidates from the KD tree. Since ring key is less informative than scan context, inspecting a small number (e.g., 10 of more than 3000 nodes) of candidates is vulnerable if there are many similar structures.

The proposed method outperformed other approaches when applied to the outdoor urban dataset. This is due to the fact the motivation for using the vertical height is from urban analysis. However, the performance is limited when applied to an indoor environment where variation in vertical height is less significant. When applied to the NCLT dataset, the scan context presented low performance both for recall and precision (left part of each graph) because the trajectory of the NCLT dataset contains narrow indoor environments where an only small area is available.

Evaluating with the Complex Urban LiDAR dataset, all methods show poorer performance than at the KITTI dataset. In particular, `Urban 02` provides the most challenging case for all methods since this sequence has narrow roads and repeated structures with similar height and rectangle shapes<sup>5</sup> compared to KITTI. The example of scan context from this challenging `Urban 02` is given in Fig. 4.6. Despite some level of performance drop is reported in this challenging dataset, the proposed method still outperformed other existing methods.

The proposed descriptor presented a strong rotation-invariance even for a reversed revisit by using view alignment based matching. For example, M2DP failed to detect a reverse loop. Among the datasets, KITTI 08 has only reverse loops and the proposed method substantially outperformed others. This phenomenon is also observed in NCLT sequences having partial reverse loops. Therefore, at NCLT sequences, M2DP reports high precision at the very low recalls because the forward loops are detected correctly. However, since reverse loops are missed, the slope of the curve rapidly decreases.

<sup>5</sup>[http://irap.kaist.ac.kr/dataset/webgl/urban02/urban02\\_sick.html](http://irap.kaist.ac.kr/dataset/webgl/urban02/urban02_sick.html)



### 4.4.3 Localization Accuracy

The proposed method can also be used when providing robust initial estimate for other localization approaches such as ICP. We conducted the experiment using KITTI 08 having reverse loops. ICP is performed point-to-point without downsampling. The example of ICP results with and without initialization are depicted in Fig. 4.8. For this sequence, we further validate the improvement in terms of both computation time and root mean square error (RMSE). Fig. 4.7 shows the improved performance with the initial yaw rotation estimates using (4.3).

### 4.4.4 Computational Complexity

Table 4.2: Average time costs on KITTI00.

	Calculating Descriptor (s)	Searching Loop (s)
Scan context-10	0.1291	0.0807
Scan context-50	0.1291	0.3331
M2DP	0.0218	0.0032
Z-projection	0.0472	0.0035
ESF	0.0635	0.0043

The average computation times evaluated on KITTI 00 are given in Table 4.2. Point cloud downsampling with a  $0.6 \text{ m}^3$  grid is used for all methods. In these experiments, the scan context creation takes longer because we employ scan context augmentation, which is non-mandatory. Thus, the time required to create a single scan context (0.0143 s, except for scan context augmentation) is shorter than it is with the other methods. The search time of the scan context includes both creation of the KD tree and computation of the distance. Scan context may require a longer search time than other global descriptors, but in a reasonable bound (2-5 Hz on Matlab).

## 4.5 Conclusion

In this chapter, we presented a spatial descriptor, *Scan Context*, summarizing a place as a matrix that explicitly describes the 2.5D structural information of an egocentric environment. Compared to existing global descriptors using a point cloud, scan context showed higher loop-detection performance across various datasets.

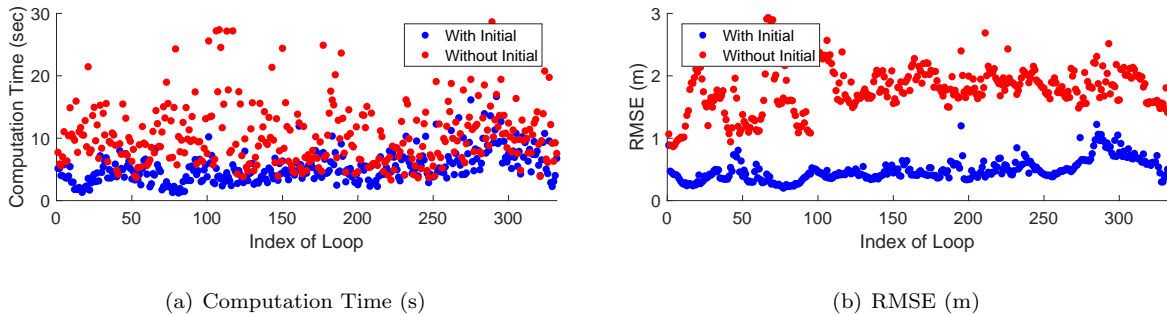


Figure 4.7: Computation time and RMSE with and without initial values. The x-axis represents the index of real loop events of KITTI 08. Blue and red indicate available and unavailable initial guesses, respectively.

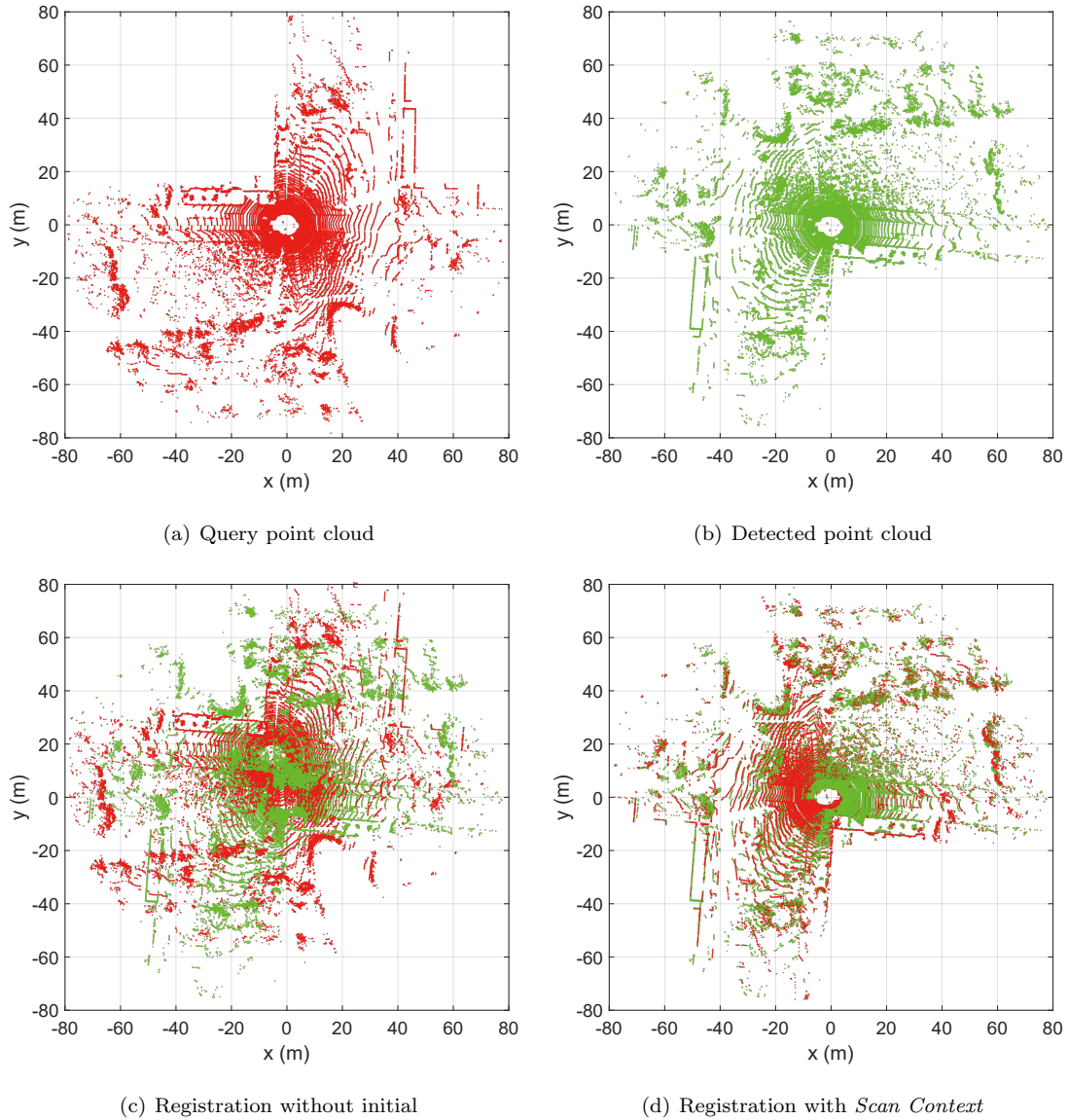


Figure 4.8: An example of point-to-point ICP results from KITTI 08. The query and the detected point clouds are from the 1785<sup>th</sup> and 109<sup>th</sup> scans, respectively. The LiDAR sensor frame represents the coordinates of the point clouds. Scoring the similarity between two scan contexts provides a coarse yaw rotation, which serves as an initial estimate to guide finer localization (i.e., ICP). In the case of this reverse loop, registration easily fails without such an initial estimate. By contrast, even this kind of unstructured environment can be registered with the use of an initial estimate obtained from the scan context.

In future work, we plan to extend scan context by introducing additional layers. That is, other bin encoding functions (e.g., a bin’s semantic information) can be used to improve performance, even for datasets with highly repetitive structures such as the Complex Urban LiDAR dataset.

## Chapter 5. Application 2: Long-term Localization

In this paper, we present a long-term localization method that effectively exploits the structural information of an environment via an image format. The proposed method presents a robust year-round localization performance even when learned in just a single day. The proposed localizer learns a point cloud descriptor, named Scan Context Image (SCI), and performs robot localization on a grid map by formulating the place recognition problem as place classification using a CNN. Our method is faster and more scalable than existing methods proposed for place recognition (e.g., [27, 70]) because it avoids a pairwise comparison between a query and scans in a database. In addition, we provide thorough validations using publicly available long-term datasets [10, 47] and show that the SCI localization attains consistent performance over a year and outperforms existing methods.

### 5.1 Introduction

Localization in a coarse [23] or fine manner [36] is one of the most necessary and basic abilities of a mobile robot. Recently, focus has moved to long-term autonomy (LTA) [61] in order to operate in a real outdoor environment beyond a lab-level static and controlled environment. LTA is particularly important for localization because the appearance of an environment changes over time (e.g., light condition or occlusion), potentially resulting in robot localization failure. Although many methods [55, 61] have been proposed, few agree on a complete visual-based solution to overcome this problem. To accomplish the LTA in changing environments, many approaches [12] have tried to take multi-experiences into a localization framework. These approaches revealed inherent drawbacks because they need to capture various conditions for the same place a priori to increase the size of the database with the number of experiences.

Contrast to visual appearance, the physical structure of a place rarely changes over time. Hence,

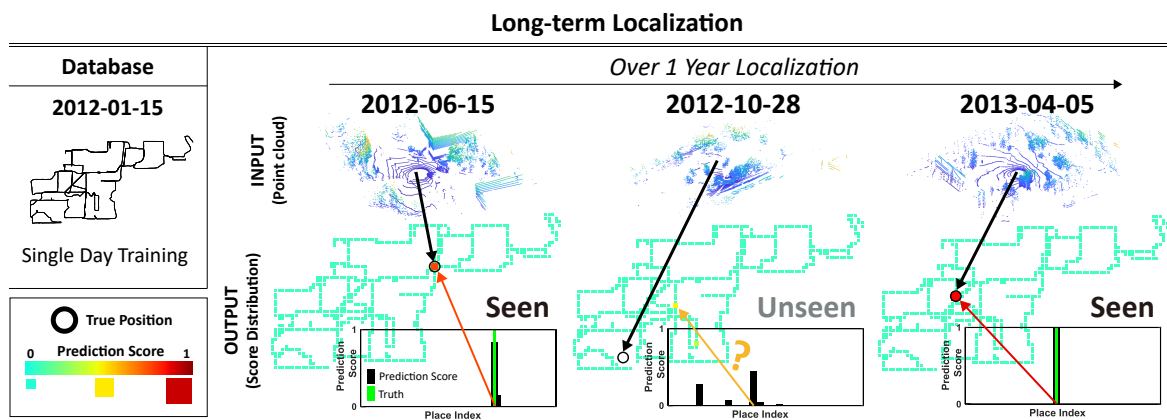
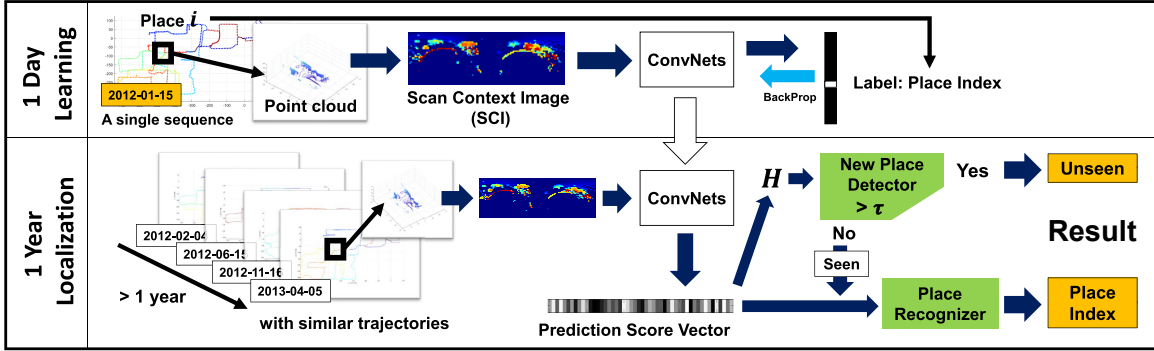


Figure 5.1: In this paper, we describe the concept of our localization method that takes only one day for a robot to learn and has consistent performance for over one year. In defining the existence of an unlearned place (i.e., an area the robot has not visited before), the algorithm we employ is capable of handling unseen places, which appear during long-term navigation.

## SCI localization Framework



## Performance Evaluation

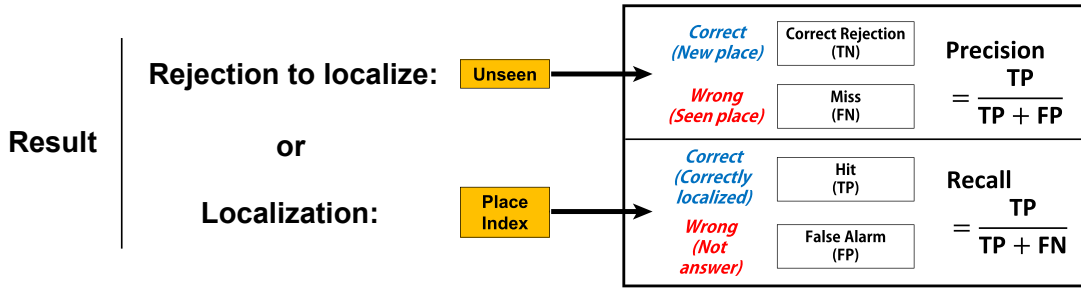


Figure 5.2: Overall pipeline of the SCI localization and performance evaluation.

leveraging structural information that is perceptible within a place has benefits for long-term localization [77] than methods based on appearance only. In that sense, a single time observation using LiDAR could represent a canonical characteristic of a place, eliminating the need for multiple experiences for robust localization. In this line of research, a handcrafted descriptor-based [27, 37] and learning-based [16, 70] method for place recognition over a point cloud has been widely proposed. However, these studies hardly captured the long-term localization requirements, including a slow but massive structural variance (e.g., construction and demolition) and unexpected viewpoint from the road topology change.

Many LiDAR-based, global, coarse localization methods have focused on making a robust descriptor with a strong capability to discriminate between places. The current research on descriptors can be divided into *non-learning* and *learning-based*.

**Non-learning based Descriptors:** M2DP [27] is a handcrafted descriptor; it projects a point cloud into multiple planes, whose normal directions are manually determined. M2DP showed that, unlike previously proposed methods such as histogram-based [74], it can effectively perform place recognition even in an outdoor context with a noisy point cloud. Inspired by the concept of 3D isovists [6] used in urban design, Scan Context (SC) [37] has shown that extracting only the highest points of a visible point cloud outperforms others including M2DP. Recently, a study on using intensity instead of structural information [13] was released.

**Learning-based Descriptors:** Recently, Uy and Lee proposed a network called PointNetVLAD [70], which combined PointNet [56] and NetVLAD [3] to generate a point cloud descriptor with achieving permutation invariance. They validated the network provided enough generality; that is, the network taught with the Oxford RobotCar [47] dataset works well for scans obtained by other robots in different

environments. Unlike PointNetVLAD’s borrowing metric learning, SegMap [16] brought an encoder-decoder system to make a descriptor into its SLAM framework so as to enable both efficient reconstruction and robust loop-closure detection.

Although many methods have been proposed, there are few empirical studies showing the effectiveness of LiDAR descriptors on long-term localization capability in urban areas. Several works have attempted to address long-term accurate (centimeter-level) localization within a prior LiDAR point cloud map using Bayesian filtering. Maddern et al. [46] proposed a 2.5D rasterized, image-based, GPU-accelerated search. Recently, Withers and Newman [73] introduced a point-wise rejection method for handling scene changes and avoiding false localization. This kind of work usually focuses on how to model structural scene changes within a Bayesian framework, rather than the global place retrieval capability of large-scale localization.

Differing from the aforementioned descriptor-based category, an end-to-end localizer that infers a robot’s pose directly using deep learning has nowadays been gaining attentions. This formulates the localization problem as 6D pose regression [36] or a coarse place classification [72]. Compared to these image-based localizers [36, 72], however, few direct methods accept a LiDAR point cloud as input have been proposed.

In this paper, we present a CNN-based, end-to-end localization framework (Fig. 5.1). The proposed localizer is based on a point cloud descriptor called Scan Context Image (SCI) that effectively summarizes the unstructured point cloud into a structured form. We validate that only a single experience is sufficient to demonstrate the effectiveness of our method on the tested datasets. Refer to the video (YouTube link, <https://youtu.be/apmmduXTnaE>) as well.

Our approach is similar to PlaNet [72] in that we also consider a place as a class and formulate a localization task as a classification task using a CNN. Unlike PlaNet, which provides a rough location scope that cannot be used for mobile robot navigation, we guarantee successful localization within a few meters on a map of a several hundred or thousand meters. Our contribution points are summarized below.

- We introduce the classification-based place retrieval pipeline using an image-shaped point cloud descriptor called SCI.
- To alleviate false alarms during long-term localization, we propose an entropy-based detection module for unseen places.
- Evaluations for two long-term datasets (the NCLT dataset [10] and the Oxford RobotCar dataset [47]) are provided. The proposed method localizes a path of over 10 km for over a year and covers all seasons and severe structural and viewpoint changes.

## 5.2 SCI Generation and Training

In this section, we introduce a 3D point cloud descriptor in an image format named SCI. Because SCI is created from a point cloud descriptor, SC, we first provide a brief review of SC. We refer readers to [37] for more detail. Next, we introduce a deep learning based classification method for long-term localization. The overall pipeline, from the training to the procedure of the localization, is depicted in Fig. 5.2.

### 5.2.1 A brief review of Scan Context (SC)

Scan Context (SC) takes a 3D point cloud as an input and divides its planar-surrounding regions within a *maximum range* into *sectors* and *rings*, which are segments divided into azimuthal and radial directions, respectively. The intersection of a sector and a ring is called a *bin*. SC only takes the highest point value from each bin and arranges them into a 2D matrix form, through which the internal arrangement of bins is preserved. The top part of Fig. 5.3 shows the making process of SC from a raw point cloud. In this paper, the number of rings, the number of sectors, and the maximum range are 40, 120, and 80 m, respectively.

### 5.2.2 Scan Context Image (SCI)

The previously defined SC is a single-channel matrix that encapsulates robust structural information (i.e., the maximum height of points) around a scene. Although SC is already in an image-like form, we normalize it and convert this into three channels to be suitable as input for CNN. When converting, the structural height out of  $[h_{min}, h_{max}]$  is saturated. In this work, we use a jet colormap, which has a larger variance than sequential colormaps, and the mapping function ( $f_c$ ) with  $h_{min} = 0$  m and  $h_{max} = 15$  m. Details are (5.1). In doing so, we empirically validate a small improvement compared with training with one channel image. The proposed SCI increases the discriminative power to more than that of SC and is also a more suitable format for inputting a CNN. This process is visualized in Fig. 5.3. We note that further investigation on network tuning for monochrome images or colormap selections may improve the localization performance.

$$c = \begin{cases} c_{\min}, & h < h_{\min} \\ f_c(h), & h \in [h_{\min}, h_{\max}] \\ c_{\max}, & h_{\max} < h, \text{ where } h \in \mathbb{R}, c \in \mathbb{R}^3. \end{cases} \quad (5.1)$$

### 5.2.3 Location Definition

Because we formulate the outdoor robot localization problem as a classification issue, we use a classification network. We first divide the region, which is covered in the training sequence, into equal-sized (e.g., 10 m by 10 m) grid cells on the x-y plane and assign a different index to each cell. A single cell represents a single place. Fig. 5.4 visualizes the concept of gridded map with 10 m cell size.

Then, all SCIs acquired in a cell are used to train a CNN with its class label; the label is a one-hot encoded vector of the corresponding place index. The label dimension is equal to the total number of places because we consider each place as a unique class. Then, the network is trained with categorical cross-entropy loss, which is generally used to train a classification network.

### 5.2.4 Network Selection

Any CNN structure (e.g., ResNet [26]) can be used to construct the proposed localization system, but we use a LeNet [43]-like network with regularization to demonstrate that our method works well with a simple network. A detailed structure of our network and parameters are shown in Table 5.1.

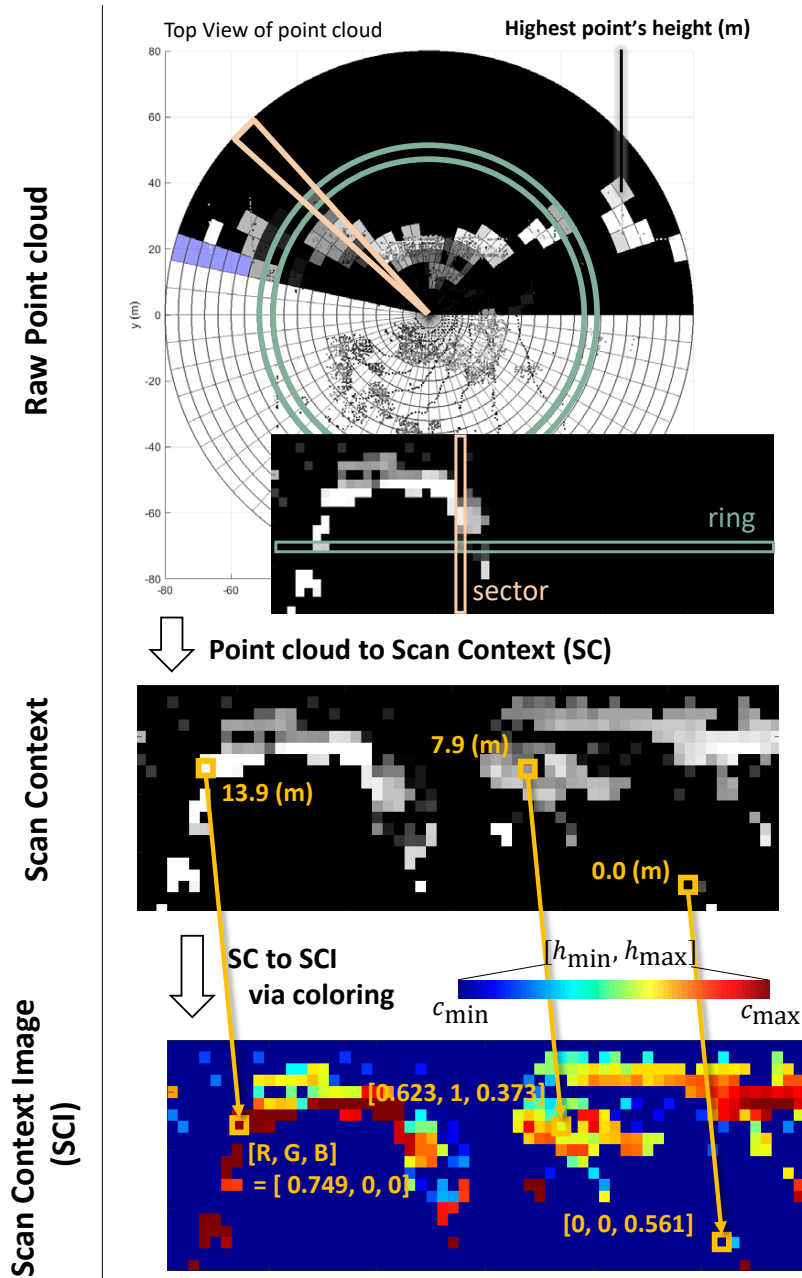
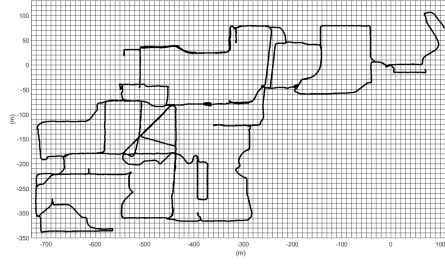


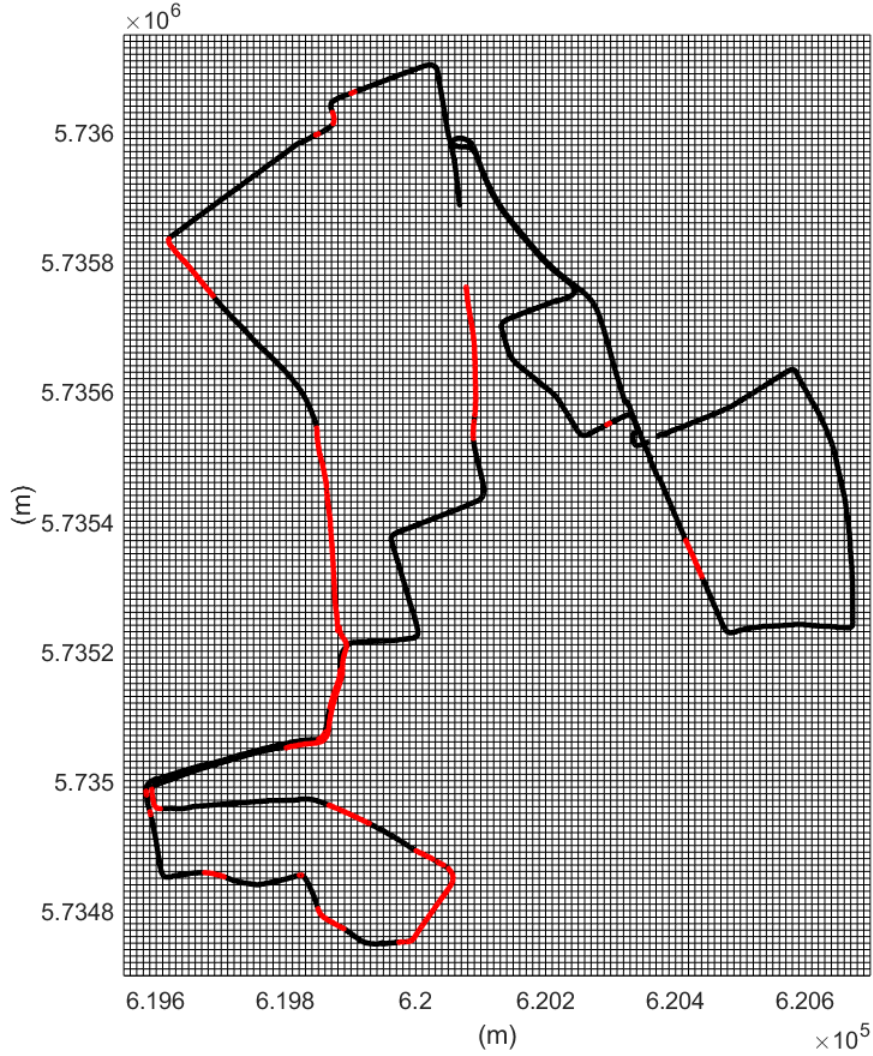
Figure 5.3: Scan Context Image (SCI) generation from a raw point cloud and conversion to a 3-channel SCI.

### 5.2.5 N-way SCI Augmentation

We propose N-way augmentation to achieve the viewpoint invariance to tackle potential viewpoint variance in the long-term localization. Because the column order of SCI indicates the heading of a robot, viewpoint variation via synthetic SCI in the training phase is fairly simple (e.g., the column-shift). Here,  $N$  is the number of 360 degrees divided by a constant interval. An example of two-way augmentation (what we call reverse augmentation) is visualized in Fig. 5.9.



(a) NCLT (2012-01-15)



(b) Oxford RobotCar (2014/07/14 15:16:36)

Figure 5.4: The example of the gridded map of the region of NCLT and Oxford RobotCar dataset. The size of grid cell here is 10 m. These trajectory is from the 2012-01-15 of NCLT and 2014/07/14 15:16:36 of Oxford RobotCar. These sequences covered 579 and 700 places, respectively. We formulate a place recognition problem as a classical deep learning-based classification problem. At the Oxford RobotCar dataset, the nodes whose INS measurements is not good (red in Fig. 5.4(b)) is excluded for the training and test.



## 5.3 SCI Localization

### 5.3.1 Un-learned Place Detection

Our objective is to learn from a single day to predict over-year place recognition. In this LTA scenario, the robot may visit a new place that is, not in the training set. Therefore, detection and proper handling of this unlearned location is critical in LTA. Prior to the localization module, we first identify whether a query place is a new place or not (i.e., a query point cloud is from a new place or not) to avoid false localization. We call the new place, *unseen place*, and an existing place in the training sequence, *seen place*. This task can be considered in unknown-unknown class detection [49], which has highly attracted computer vision researchers for constructing more robust classification system. For example, Dropout Variational Inference [35] can approximately provide a class probability but requires multiple predictions, which is time consuming and thus may be difficult for real-time robot localization.

Unlike this costly method, we propose a way to directly use the entropy of the output vector (without dropout at the test time) from the network. Note that we do not aim to approximate each class probability; instead, we rather focus on identifying whether the query is seen or unseen. As will be shown in §5.5, this entropy of the output vector has a substantially stronger discriminative performance than traditional distance-based thresholding. Specifically, we use the following normalized entropy of the prediction score vector

$$H(\mathbf{p}) = -\frac{1}{H_{max}} \sum_{i=1}^N p_i \log_2 p_i, \quad (5.2)$$

where  $p_i$  is the  $i^{\text{th}}$  element of the vector  $\mathbf{p}$  and  $H_{max}$ , which is the maximum entropy of a  $N$  dimensional vector, exists for the normalization.

If the entropy of the prediction vector is higher than a given threshold  $\tau$  (user parameter), it is considered as a new place and rejected without localization. On the other hand, we only perform localization in the following step for images classified as *seen*. Fig. 5.5 shows an example of the distribution of entropies from seen and unseen places of the sequence.

Table 5.1: A simple network structure we used. BN and MP are batch normalization and max pooling, respectively, and we used  $2 \times 2$  pooling size. The number in the Conv() and FullyConnected() layer means the number of filters and the number of nodes, respectively.  $5 \times 5$  filters were used for all convnets and 0.7 (30 % remains) dropouts were applied for all Dropout layers.  $N$  is the number of total places. We trained the network with 64 of batch size and using Adam optimizer with default parameters (learning rate = 0.001,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ).

<b>Input</b>	(batch_size, 40, 120, 3 )
Conv1	BN(MP(ReLU(Conv(64, Input))))
Conv2	BN(MP(ReLU(Conv(128, Conv1))))
Conv3	Flatten(MP(ReLU(Conv(256, Conv2))))
FC1	FullyConnected(64, Dropout(Conv3))
FC2	softmax(FullyConnected( $N$ , Dropout(FC1)))
<b>Output</b>	(batch_size, $N$ )

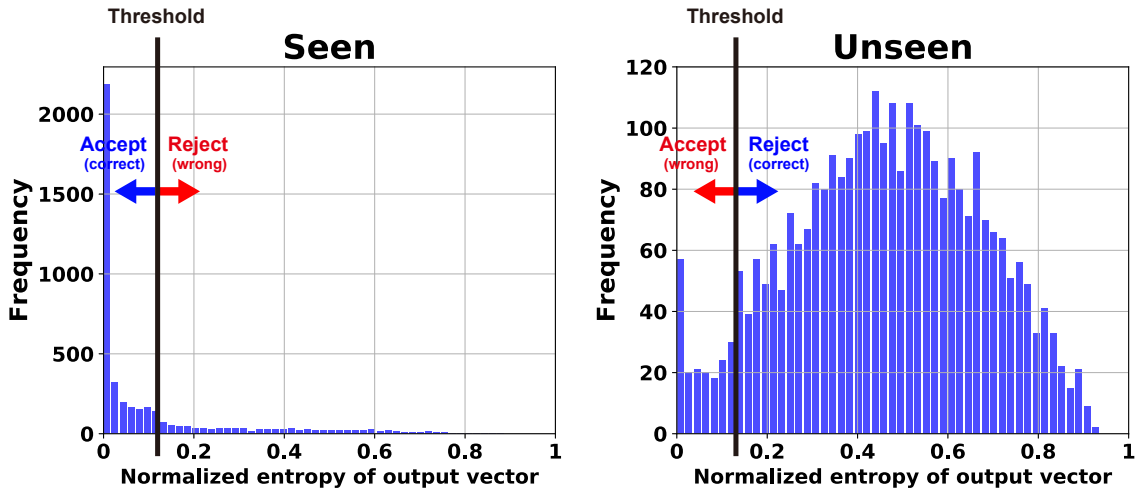


Figure 5.5: The example of the distribution of entropies of prediction score vectors for seen and unseen. This example is from the test sequence 2015-06-12-08-52-55 of the Oxford RobotCar dataset. Left: The histogram of predictions’ entropies from *seen* places usually has small values. Right: The histogram of entropies from *unseen* (*new*) places has a large variance, and there are higher entropies than the case of seen places.

### 5.3.2 Localization

If the query is considered to be seen (i.e., the point cloud is obtained from seen places), then localization is performed using the prediction score vector. The index of this vector’s element, which has the largest score, is concluded as the current place. More generally, we would say the localization is successful if the ground truth index of a query place belongs to a set of top  $N$  indexes whose scores are in a larger order in the network’s prediction score vector.

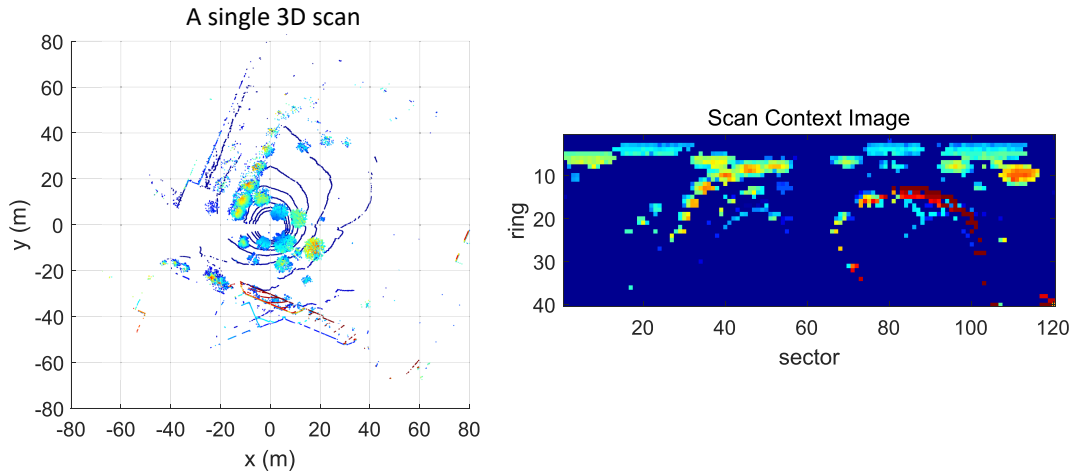
## 5.4 Experiments

In this section, we first describe experimental settings and datasets used for evaluation.

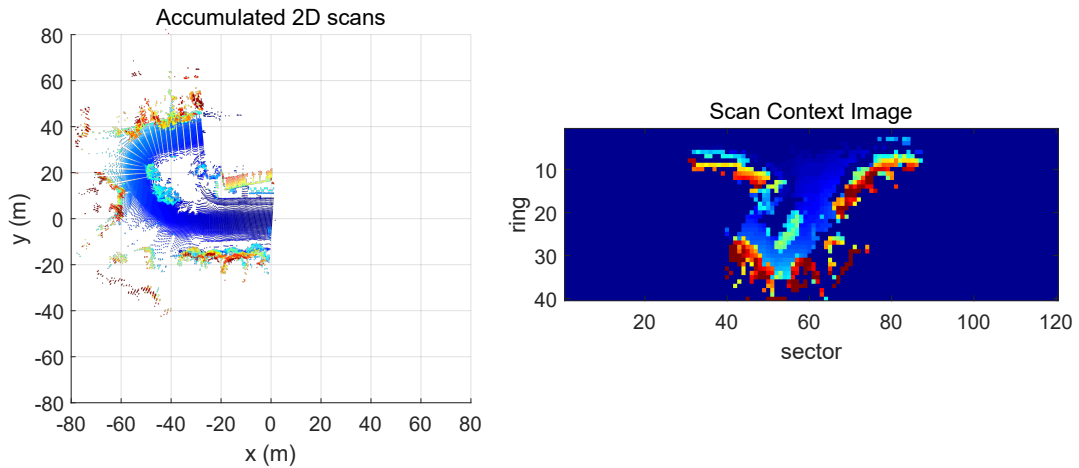
### 5.4.1 Benchmark Datasets

We used two long-term datasets that are publicly available in the robotics community: the NCLT [10] and Oxford RobotCar [47] datasets. Both datasets provide multiple sequences along similar trajectories over a year and include various environmental changes for the same places.

The NCLT dataset provides 3D LiDAR scans and each scan is directly encoded into the SCI as described on the left of Fig. 5.6. For the Oxford RobotCar dataset, we used sequences with the *full* trajectory of nearly 10 km. This dataset has no 3D LiDAR, and 2D LiDAR were mounted perpendicularly to the vehicle’s moving direction. Thus we accumulated 2D scans along a local trajectory for enough length as visualized on the right of Fig. 5.6. We set an accumulation length (or a window size) equal to the *maximum range*, which is the parameter of an SCI. We use the visual odometry the dataset provides for stacking scans. By stacking them, we use the relative motion between a previous scan and a recent scan is placed at the origin. In doing so, we can make a 3D point cloud (or a submap) with enough



(a) Sample point cloud and SCI from NCLT dataset



(b) Sample point cloud and SCI from Oxford RobotCar dataset

Figure 5.6: Visualization of a point cloud and the associated SCI for (a) 3D LiDAR and (b) 2D LiDAR. In (b), the accumulated point cloud depends on the trajectory and may result in missing information. Despite this missing portion in the SCI, the deep-learning-based localizer successfully matched the place from the trained map.

information to make an SCI. We considered the global coordinate available from the *ins.csv* file as the ground truth of each place. Only places with a reliable inertial navigation system (INS) status (i.e., `INS_SOLUTION_GOOD`) are used for training and tests.

The size of a grid cell for the main analysis (Fig. 5.7 and Fig. 5.8) is 10 m by 10 m. For this grid map resolution, the NCLT and Oxford RobotCar datasets have 579 and 700 places in their training sequences, respectively. The places from the NCLT and Oxford RobotCar datasets are trained with only a single sequence, and then the localization is evaluated for the following 10 sequences, which covers over a year. For training, an SCI and descriptors of comparison methods are sampled for every 1 m. The test sequences are also evaluated by sampling every 1 m. Details about the training and test sequences of the NCLT and Oxford RobotCar datasets are summarized in Table 5.2. The seen and unseen rows indicates the number of queries from seen and unseen places.

## 5.4.2 Comparison Methods

We compare our method, SCI-localization, with three state-of-the-art handcrafted and learning-based point cloud descriptors: M2DP [27], Scan Context [37], and PointNetVLAD [70]. For a fair comparison, both methods construct a database using only descriptors from a single sequence and are compared to a query descriptor from test sequences for over a year. The nearest candidate’s index is considered a query’s location.

*M2DP* is a lightweight point cloud descriptor designed for loop-closure detection. The core idea of M2DP is projecting a 3D point cloud into multiple 2D planes. We used the same parameters and procedure as the original authors by using the open-source of M2DP<sup>1</sup>, and we acquire a 192-dimension descriptor from a point cloud.

*Scan Context-50* exploits a similar descriptor as the SCI, but in a non-learning based way. SC used the column-wise comparison to calculate the distance between a query and a candidate. To clearly validate the learning effect, we compare SCI against Scan Context-50 in [37], which is the method that takes 50 candidates for the pairwise comparison.

*PointNetVLAD* is a combination of PointNet [56] and NetVLAD [3], so it can directly consume a point cloud without any reformulation such as projection or voxelization. We applied preprocessing similar to the original paper; a ground-removed point cloud within a  $[-25\text{ m}, 25\text{ m}]$  cubic window is filtered into the constant number (4096) points and rescaled into a  $[-1, 1]$  range with a zero mean. This processed point cloud is fed to the network, and, finally, we get a 256-dimensional descriptor. We used the pretrained model (refined version) the authors provided<sup>2</sup>.

## 5.5 Evaluation Results

In this section, we provide intensive analyses to validate the effectiveness and robustness of the proposed method. The detailed information of training data and test sequences are described in Table 5.2. The test sequences of each dataset were possibly selected to include at least one sequence per month to cover various conditions over the entire year. The number of samples in rows of seen and unseen places in Table 5.2 are sampled per every 1 m and are used for the evaluation.

### 5.5.1 Precision-recall Curve

We first evaluate the general performance using precision-recall curve for both datasets throughout the long-term operation (Fig. 5.7). The evaluation procedure is depicted in the right side in Fig. 5.2. We

<sup>1</sup><https://github.com/LiHeUA/M2DP>

<sup>2</sup><https://github.com/mikacuy/pointnetvlad>

Table 5.2: Summary of datasets

Dataset	Train Seq.	Test Seq.										
		2012-01-15	2012-02-04	2012-03-17	2012-05-26	2012-06-15	2012-08-20	2012-09-28	2012-10-28	2012-11-16	2013-02-23	2013-04-05
NCLT	579 places	Seen	5170	5449	5533	3321	5146	4626	4623	3575	4114	3341
		Unseen	441	428	773	742	835	919	1034	1290	1095	1162
Oxford Robot	2014-07-14 -14-49-50	2014-07-14	2014-11-25	2014-12-17	2015-02-03	2015-03-10	2015-04-17	2015-05-22	2015-06-12	2015-07-10	2015-08-13	
		-15-16-36	-09-18-32	-18-18-43	-08-45-10	-14-18-10	-09-06-25	-11-14-30	-08-52-55	-10-01-59	-16-02-58	
Car	700 places	Seen	4079	5484	3926	5657	5106	5485	5664	4321	4872	5043
		Unseen	414	2066	1769	2163	2782	2571	2232	3062	2585	2466

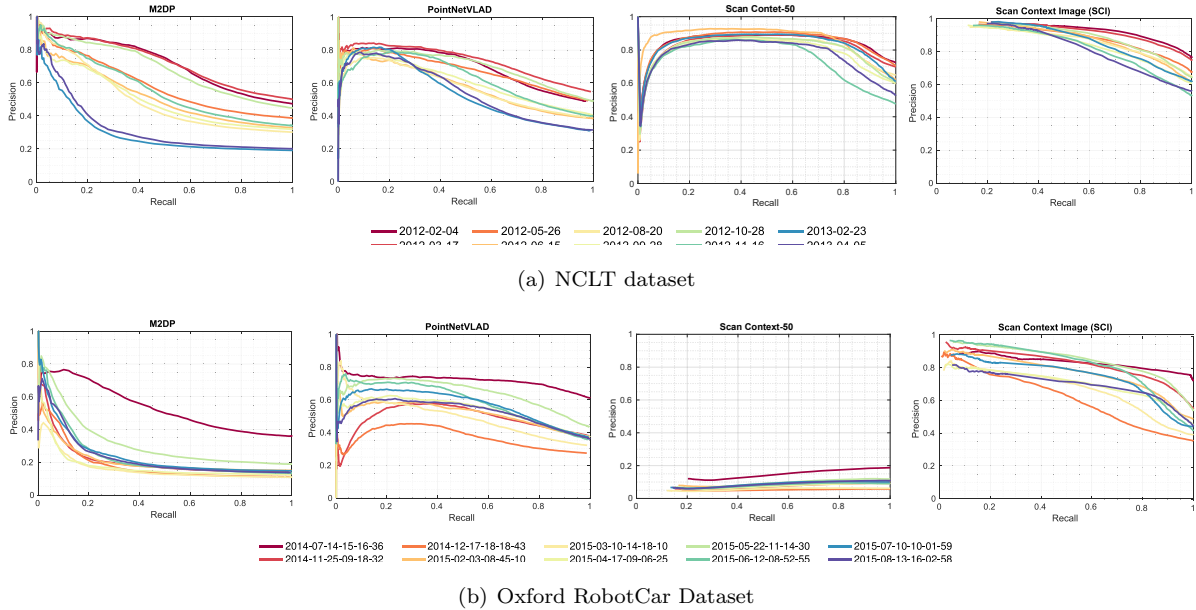


Figure 5.7: Precision-recall curves for two long-term datasets, NCLT and Oxford RobotCar dataset.

considered the localization to be correct if the index of the largest element of the network’s output vector (for our method) or nearest descriptor’s place index (for M2DP, Scan Context-50 and PointNetVLAD) is the same as the answer (i.e., top 1 performance).

The learning-based descriptor, PointNetVLAD, outperformed the handcrafted method, M2DP, by a large margin. However, PointNetVLAD revealed lower performance than our method in terms of long-term localization performance. Moreover, the proposed SCI localization method presented less fluctuation than others in performance over time. For Scan Context-50, the column-wise matching function of SC assumes a surround-capturing LiDAR; thus it showed the poor performance at the Oxford RobotCar dataset, which used 2D LiDAR. SCI decreases performance over time but still performs better than other methods.

### 5.5.2 Retrieval Capability

For large-scale localization, not only the top 1, but taking more candidates (e.g., top 5 and top 25) would also be meaningful. Therefore, we provide a more in-depth analysis of the retrieval power of each method. We extended the criteria of the correct answer to the top 5 and top 25 candidates to investigate by how much the performance of each method increased.

Fig. 5.8 shows a comparison of overall performance. We plot the area under the curve (AUC) of the precision-recall curve of each sequence as a measure. The closer the AUC is to 1, the more perfect the localization. The AUC values of all methods have increased by allowing the top 25 candidates but our top 1 performance is comparable or better than others’ top 25 performance.

### 5.5.3 Long-term Robustness

In this subsection, we investigate two types of environmental changes; *non-structural* and *structural* changes.

**Non-structural changes:** Although the structural information of a scene is naturally robust for LTA, there are a few challenging factors that make a point cloud different from the experience. Fig. 5.9

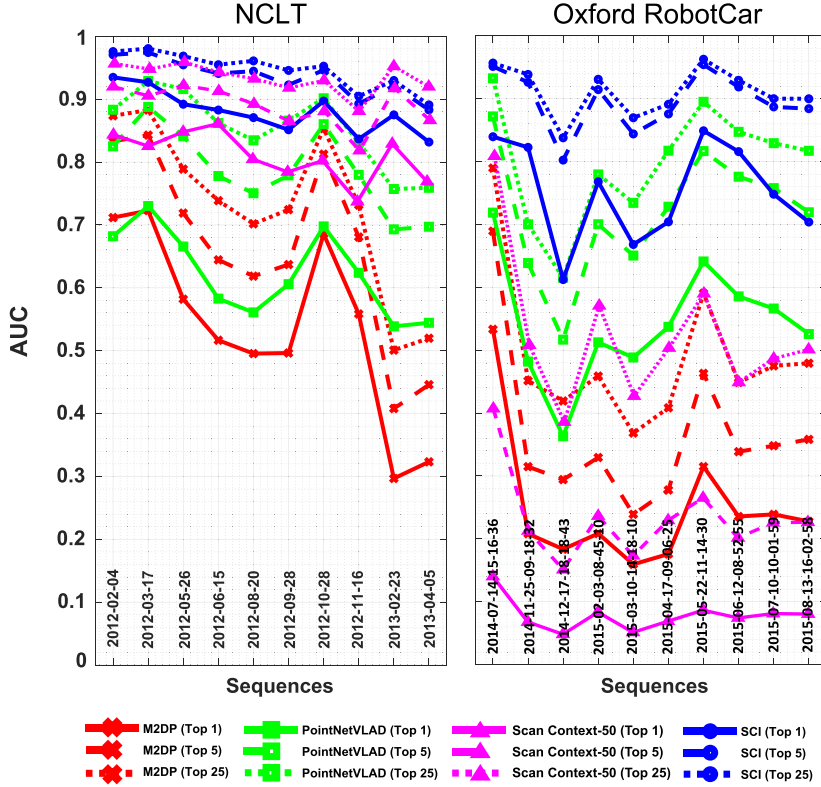


Figure 5.8: AUC performance changes over time for different criteria of success localization.

visualizes the examples of challenging cases and their corresponding SCIs from the NCLT dataset. The NCLT dataset always had partial and varying occlusion due to an accompanying human pilot. In addition, the Segway-like robot used in the NCLT dataset had an unstable roll motion compared to a car platform, and partial structures such as foliage changed over time. Despite these challenging factors, our method successfully localized a query because the SCI preserved the internal relations of the egocentric scene structure, unlike the other descriptors that lost the original scene’s structural shapes.

**Structural changes:** The long-term structural challenges arise from structural experience (i.e., structures that existed at the training sequence) that may have disappeared (*demolition*) or been newly constructed (*construction*) over time. For validation, we removed points within a randomly selected sector or added new randomly generated wall-shaped points, as in Fig. 5.10(a). Because the M2DP is based on point projection, it is less affected by the appearance of structures but is vulnerable to demolition. PointNetVLAD was sensitive to the removal and addition of points as it uses only 4096 points as the input. Although SC utilizes descriptors very similar to SCI, we verified that ConvNet based unseen place detection and classification-based retrieval are superior in localization performance.

### 5.5.4 Robustness to Viewpoint Changes

Arbitrary viewpoint variation inevitably occurs during the long-term localization. In this section, we examine the effect of N-way augmentation on the viewpoint change robustness by increasing N in the training phase. Unfortunately, the number of queries in original datasets are rather small for testing various viewpoint cases. Instead, we tested the trained network by randomly rotating a query point cloud’s heading.

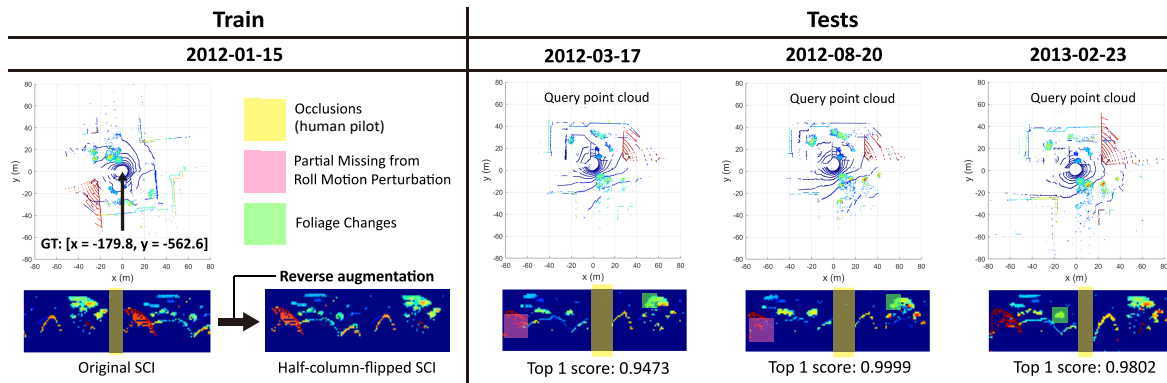
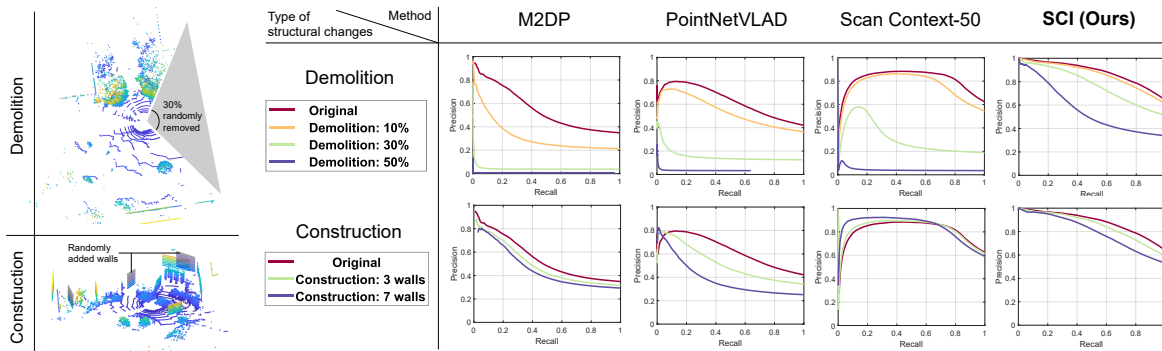


Figure 5.9: Robustness to non-structural changes. Despite challenging factors (e.g., viewpoint changes, occlusions, and foliage), the proposed method successfully found its location with a high score for over hundreds of places over a year.



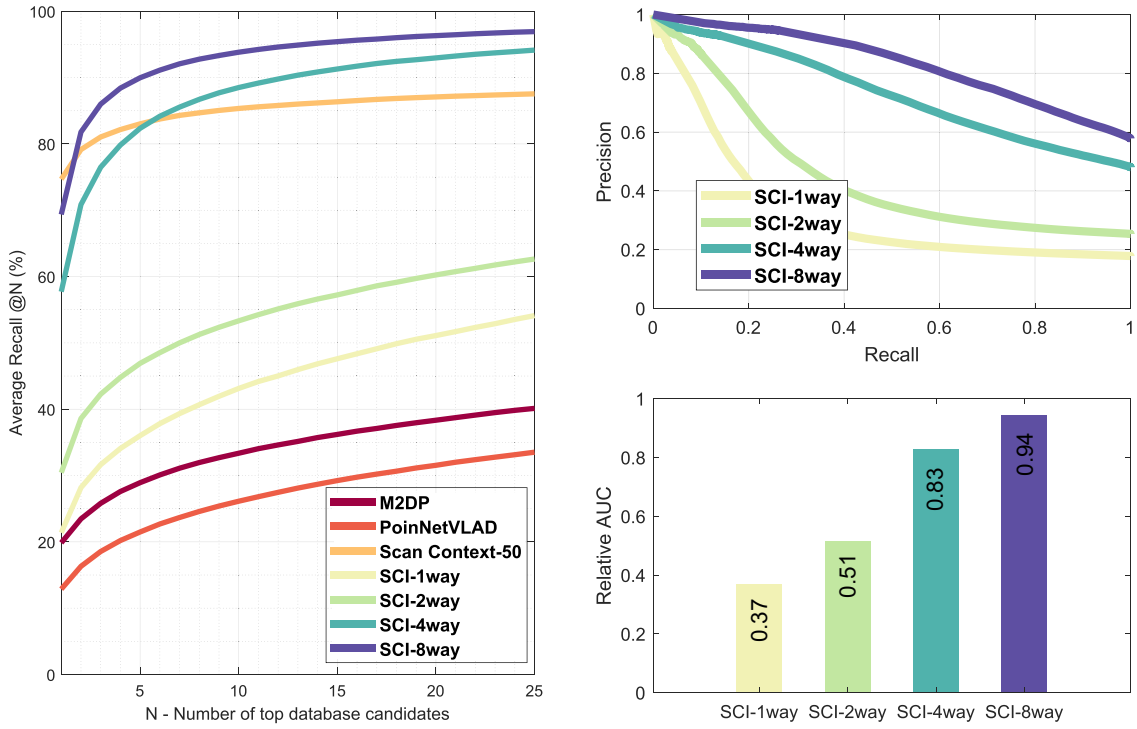
(a) Examples of structural changes (b) Precision-recall curves for demolition and construction. Each line is a mean for 10 test sequences.

Figure 5.10: Robustness to structural changes on the test sequences of the NCLT dataset.

As seen in Fig. 5.11(a), existing descriptors, including the baseline of SCI-localization without N-way augmentation, failed to localize under arbitrary viewpoint changes. We empirically validated that using four-way augmentation could yield sufficient robustness to the viewpoint variation. In doing so, the general performance is also preserved as in Fig. 5.11(b). The bottom of Fig. 5.11(b) presents the AUC relative to the original performance in Fig. 5.7(a) without the intentional heading rotation.

### 5.5.5 Grid Cell Size

We also evaluated localization performance by considering different grid sizes to identify whether finer localization is possible. We conducted the same experiment as in §5.5.1 but with different grid cell sizes. The grid cells are finer (5 m by 5 m) and coarser (20 m by 20 m). The number of output nodes in the SCI localization network was reset to the new total number of places and retrained. The results for different grid cell sizes are shown in Fig. 5.12. Despite increased labels of over 1000 places for 5 m<sup>2</sup> resolution and a slight decrease in performance, our method still presented higher performance with lower variance than PointNetVLAD with a 10 m resolution for both datasets.



(a) Retrieval performance of each method for random viewpoint changes (b) Precision-recall curves for each N-way augmentation.

Figure 5.11: Robustness to random viewpoint changes on the test sequences of the NCLT dataset. Each line is a mean for 10 test sequences.

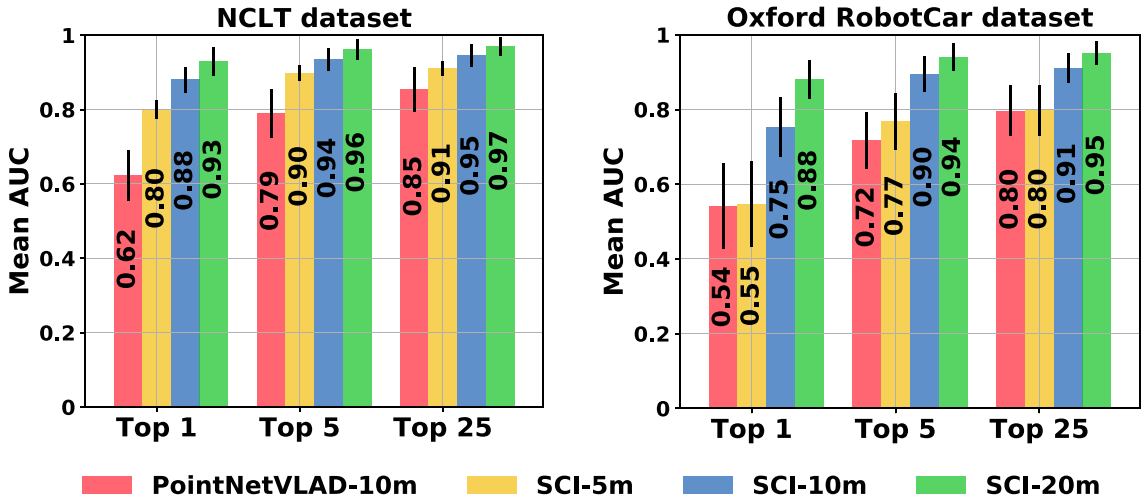


Figure 5.12: Performances with respect to different grid cell sizes. The vertical black line pinned at each bar represents the standard deviation of all (10 for each dataset) test sequences.

### 5.5.6 Runtime Evaluation

Another strength of the proposed method is the lightweight implementation. For the runtime comparison in Table 5.3, all implementations used Matlab except for a few parts that used the deep network



Table 5.3: Average time cost for each methods. The comparison is conducted on the 2013-04-05 of the NCLT dataset.

<b>Method</b>	Descriptor Generation (sec)	Retrieval (sec)	Total (sec)
SCI	0.0434	0.0047	0.0481
SC-50	0.0413	0.4633	0.5046
PNVLAD	0.1374	0.0220	0.1594
M2DP	0.0758	0.0195	0.0953

by using Python at NVIDIA GTX 1080Ti with a test batch size of one.

PointNetVLAD showed the longest time for a generation, requiring both preprocessing (e.g., ground removal and filtering) and passing the network. However, both PointNetVLAD and M2DP are lightweight descriptors, and thus find a nearest in the database quickly (i.e., short retrieval time). Scan Context-50 is the slowest for retrieval, as reported in [37]. Unlike other methods, the SCI’s retrieval time is the shortest because SCI-localization directly obtains scores for  $N$  places via a single pass through the network rather than a pairwise comparison with the whole database.

## 5.6 Conclusion

We presented a global end-to-end localization method based on deep learning by learning the novel point cloud descriptor, SCI. The proposed SCI with a classification network is more robust to the long-term robot localization of other state-of-the-art pairwise scoring-based place retrieval methods [27, 70]. We conducted extensive evaluations on public long-term datasets (NCLT and Oxford RobotCar), and our method showed a consistent, and state-of-the-art performance for over a year even though the network was trained using only a single sequence. Due to its robust and global performance, we expect the proposed framework could also be used for the kidnapped robot problem.

## Chapter 6. Conclusion

Overall summary of this thesis and future works are provided in this chapter.

### 6.1 Contributions

Through this thesis, we tried to answer the single question, ‘What is the sense of a place and where does the sense of place originate from?’. We found a clue for this answer from the *isovist*, which was originally invented from the space analysis researches, and proposed a novel point cloud descriptor called a Scan Context, which is inspired from the isovist. The detail consideration was provided in chapter §3.

We have empirically showed that this descriptor with the meaning of 3D isovist shows robust performances in large gap compared to existing methods in robot localization.

Therefore, we would like to say from these promising results: The isovist of the place, i.e., the openness of the place, is important not only to a human but also to a robot. Although a place can be defined in various ways (e.g., set of semantic objects or visual appreciation), openness is the one of the powerful measure that summarizes the unique characteristics of a place, which is distinguishing it from other places.

### 6.2 Future Work

In this subsection, we discuss some feasible ideas, which are remained for future works.

- Learning-based Scan Context Similarity Function. Although Scan Context is robust, there is a limitation that the column-wise comparison (4.2) for online place recognition is still slow. Therefore, we will propose a deep network-based method to measure the similarity between two Scan Contexts.
- Flexible Grid Map. For long-term localization, we proposed the predefined grid map-based localization using the classification network. The fixed-size of grids makes it is hard to recognize a new but actually near to a seen place that is an adjacent grid of a seen grid, which exists in a training sequence. Therefore, we need to devise a more flexible map representation.
- Lidar Deep Odometry. We showed that rotation registration is possible using Scan Context. We did not model translation in this thesis, but we expect translation to be inferred as well, particularly with help of deep learning. Therefore, if we can infer the relative pose between the two Scan Contexts, we expect that a full deep learning based LiDAR odometry would be possible.
- Scan context generation from image. A LiDAR sensor has been pointed out that unlike images, it provides rich structure information without being affected by light conditions, but it is nevertheless more expensive than a camera, thus the LiDAR sensor is hard to equipped for every autonomous robot. Therefore, it is very natural question whether we can generate a Scan Context from an image. In other words, it is a study of whether to produce isovist and openness information of place from image.

- 3D Isovist meets Robot Perception: robotics-aided realistic, real-time, and large-scale urban Analysis. Scan Context effectively summarizes the openness of a place. In other words, it could mean real-time large-scale urban analysis can be performed by an autonomous robot equipped with a LiDAR and using Scan Context. For the conventional methods in the urban design field, it was necessary to construct a model in order to analyze the openness of city-size scale. However, using LiDAR and Scan Context, the robot can analyze the urban space at the same time of driving in the city without any model and offline analysis.

## Bibliography

- [1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. <http://ceres-solver.org>.
- [2] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008.
- [3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.
- [4] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.
- [5] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [6] Michael L Benedikt. To take hold of space: isovists and isovist fields. *Environment and Planning B: Planning and design*, 6(1):47–65, 1979.
- [7] Michael Bosse and Robert Zlot. Keypoint design and evaluation for place recognition in 2D lidar maps. *Robotics and Autonomous Systems*, 57(12):1211–1224, 2009.
- [8] Michael Bosse and Robert Zlot. Place recognition using keypoint voting in large 3d lidar datasets. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2677–2684. IEEE, 2013.
- [9] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.
- [10] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of Michigan North Campus Long-Term Vision and Lidar Dataset. *International Journal of Robotics Research*, 35(9):1023–1035, 2016.
- [11] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956, 2018.
- [12] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *International Journal of Robotics Research*, 32(14):1645–1661, 2013.
- [13] Konrad Cop, Paulo Vinicius Koerich Borges, and Renaud Dubé. DELIGHT: An Efficient Descriptor for Global Localisation using LiDAR Intensities. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018. In press.
- [14] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *International Journal of Robotics Research*, 27(6):647–665, 2008.

- [15] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMatch: Segment based place recognition in 3D point clouds. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5266–5272, 2017.
- [16] Renaud Dubé, Andrei Cramariuc, Daniel Dugas, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMap: 3D Segment Mapping using Data-Driven Descriptors. *Proceedings of the Robotics: Science & Systems Conference*, 2018.
- [17] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.
- [18] Hugh F Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal on Robotics and Automation*, 4(1):23–31, 1988.
- [19] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.
- [20] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.
- [21] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.
- [22] Dorian Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2197158.
- [23] D. Galvez-López and J. D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.
- [24] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.
- [25] Fei Han, Xue Yang, Yiming Deng, Mark Rentschler, Dejun Yang, and Hao Zhang. Sral: Shared representative appearance learning for long-term visual place recognition. *IEEE Robotics and Automation Letters*, 2(2):1172–1179, 2017.
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [27] Li He, Xiaolong Wang, and Hong Zhang. M2DP: a novel 3D point cloud descriptor and its application in loop closure detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 231–237, 2016.
- [28] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1271–1278. IEEE, 2016.

- [29] Marian Himstedt, Jan Frost, Sven Hellbach, Hans-Joachim Böhme, and Erik Maehle. Large scale place recognition in 2D LIDAR scans using geometrical landmark relations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5030–5035, 2014.
- [30] Jinyong Jeong, Younggun Cho, and Ayoung Kim. Road-SLAM : Road marking based SLAM with lane-level accuracy. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 1736–1473, Redondo Beach, CA, Jun. 2017.
- [31] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. Complex Urban LiDAR data set. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018.
- [32] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.
- [33] M. Kaess. *Incremental Smoothing and Mapping*. Ph.D., Georgia Institute of Technology, Dec 2008.
- [34] Fabjan Kallasi and Dario Lodi Rizzini. Efficient loop closure based on FALKO lidar features for online robot localization and mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1206–1213, 2016.
- [35] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems Conference*, pages 5574–5584, 2017.
- [36] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2938–2946, 2015.
- [37] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Oct. 2018. Accepted. To appear.
- [38] Hyungjin Kim, Bingbing Liu, Chi Yuan Goh, Serin Lee, and Hyun Myung. Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area. *IEEE Robotics and Automation Letters*, 2(3):1518–1524, 2017.
- [39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems Conference*, 2012.
- [40] Rainer Kümmeler. State estimation and optimization for mobile robot navigation. *Universität Freiburg*, 2013.
- [41] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomáš Krajník. Artificial intelligence for long-term robot autonomy: A survey. *arXiv preprint arXiv:1807.05196*, 2018.
- [42] Yasir Latif, César Cadena, and José Neira. Robust loop closing over time for pose graph SLAM. *International Journal of Robotics Research*, 32(14):1611–1626, 2013.

- [43] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [44] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.
- [45] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.
- [46] W. Maddern, G. Pascoe, and P. Newman. Leveraging experience for large-scale LIDAR localisation in changing cities. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1684–1691, 2015.
- [47] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15, 2017.
- [48] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1643–1649, 2012.
- [49] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout Sampling for Robust Object Detection in Open-Set Conditions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018. In press.
- [50] Eugenio Morello and Carlo Ratti. A digital image of the city: 3d isovists in lynch’s urban analysis. *Environment and Planning B: Planning and Design*, 36(5):837–853, 2009.
- [51] Naveed Muhammad and Simon Lacroix. Loop closure detection using small-sized signatures from 3D LIDAR data. In *Safety, Security, and Rescue Robot. (SSRR), IEEE Intl. Symp. on*, pages 333–338, 2011.
- [52] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [53] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 2564–2570, 2014.
- [54] Edwin B Olson. Robust and efficient robotic mapping. 2008.
- [55] Horia Porav, Will Maddern, and Paul Newman. Adversarial Training for Adverse Conditions: Robust Metric Localisation using Appearance Transfer. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018. In press.
- [56] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. pages 77–85, 2017.
- [57] Dario Lodi Rizzini. Place recognition of 3D landmarks based on geometric relations. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

- [58] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3384–3391, 2008.
- [59] Samuele Salti, Federico Tombari, and Luigi Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.
- [60] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011.
- [61] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic Visual Localization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. In press.
- [62] Young-Sik Shin, Yeong Sang Park, and Ayoung Kim. Direct visual slam using sparse depth for camera-lidar system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Brisbane, May. 2018. In print.
- [63] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.
- [64] Randall C Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68, 1986.
- [65] Cyrill Stachniss. *Exploration and mapping with mobile robots*. PhD thesis, University of Freiburg, 2006.
- [66] Bastian Steder, Michael Ruhnke, Slawomir Grzonka, and Wolfram Burgard. Place recognition in 3D scans using a combination of bag of words and point feature based relative pose estimation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1249–1255, 2011.
- [67] Niko Sünderhauf. *Robust optimization for simultaneous localization and mapping*. PhD thesis, Technischen Universität Chemnitz, 2012.
- [68] Niko Sünderhauf and Peter Protzel. BRIEF-Gist-Closing the loop by simple means. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1234–1241, 2011.
- [69] Georgi Tinchev, Simona Nobili, and Maurice Fallon. Seeing the wood for the trees: reliable localization in urban and natural environments. *arXiv preprint arXiv:1809.02846*, 2018.
- [70] Mikaela Angelina Uy and Gim Hee Lee. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. In press.
- [71] Christoffer Valgren and Achim J Lilienthal. SIFT, SURF and Seasons: Long-term outdoor localization using local features. In *European Conference on Mobile Robotics*, pages 253–258, 2007.



- [72] Tobias Weyand, Ilya Kostrikov, and James Philbin. PlaNet - photo geolocation with convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, pages 37–55, 2016.
- [73] D. Withers and P. Newman. Modelling Scene Change for Large-Scale Long Term Laser Localisation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6233–6239, 2017.
- [74] Walter Wohlkinger and Markus Vincze. Ensemble of shape functions for 3d object classification. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 2987–2992. IEEE, 2011.
- [75] Ryan W Wolcott and Ryan M Eustice. Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving. *The International Journal of Robotics Research*, 36(3):292–319, 2017.
- [76] Perry Pei-Ju Yang, Simon Yunuar Putra, and Wenjing Li. Viewsphere: a gis-based 3d visibility analysis for urban design evaluation. *Environment and Planning B: Planning and Design*, 34(6): 971–992, 2007.
- [77] Yawei Ye, Titus Cieslewski, Antonio Loquercio, and Davide Scaramuzza. Place Recognition in Semi-Dense Maps: Geometric and Learning-Based Approaches. *Proceedings of the British Machine Vision Conference*, 2017.
- [78] Jinyong Jeong Youngji Kim and Ayoung Kim. Stereo camera localization in 3D LiDAR maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Oct. 2018. Accepted. To appear.
- [79] Hao Zhang, Fei Han, and Hua Wang. Robust multimodal sequence-based loop closure detection via structured sparsity. In *Proceedings of the Robotics: Science & Systems Conference*, 2016.
- [80] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, page 9, 2014.