Contributions to the use of 3D lidars for autonomous navigation: calibration and qualitative localization

Naveed Muhammad

To cite this version:

Naveed Muhammad. Contributions to the use of 3D lidars for autonomous navigation: calibration and qualitative localization. Robotics [cs.RO]. INSA de Toulouse, 2012. English. NNT : . tel-00676797

HAL Id: tel-00676797
https://theses.hal.science/tel-00676797

Submitted on 6 Mar 2012

Université
de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

Discipline ou spécialité :

Robotique

Présentée et soutenue par :
Naveed MUHAMMAD
le : mercredi 1 février 2012

Titre :

Contributions to the use of 3D lidars for autonomous navigation: calibration and qualitative localization

---

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
LAAS-CNRS

Directeur(s) de Thèse :

Simon LACROIX

Rapporteurs :

David FOFI
Fawzi NASHASHIBI

Membre(s) du jury :

Rachid ALAMI
Roland CHAPUIS
Paul CHAVENT

Abstract

In order to autonomously navigate in an environment, a robot has to perceive its environment correctly. Rich perception information from the environment enables the robot to perform tasks like avoiding obstacles, building terrain maps, and localizing itself. Classically, outdoor robots have perceived their environment using vision or 2D lidar sensors. The introduction of novel 3D lidar sensors such as the Velodyne device has enabled the robots to rapidly acquire rich 3D data about their surroundings. These novel sensors call for the development of techniques that efficiently exploit their capabilities for autonomous navigation.

The first part of this thesis presents a technique for the calibration of 3D lidar devices. The calibration technique is based on the comparison of acquired 3D lidar data to a ground truth model in order to estimate the optimal values of the calibration parameters. The second part of the thesis presents a technique for qualitative localization and loop closure detection for autonomous mobile robots, by extracting and indexing small-sized signatures from 3D lidar data. The signatures are based on histograms of local surface normal information that is efficiently extracted from the lidar data. Experimental results illustrate the developments throughout the manuscript.

To my parents, Maqbool and Shabana

Acknowledgements

I am very thankful to my advisor Dr. Simon Lacroix for his guidance, ideas, encouragement and patience throughout the course of my Ph.D. He was always there to listen to me and help me with any issues, technical or otherwise.

I'm also very thankful to the RIA group at LAAS for providing me the opportunity to carry out my Ph.D. research here. I'd like to thank DGA and ANR for funding my research under the Action and 2RT3D projects respectively. I'd like to thank Matthieu for providing his technical help, whenever I needed it, for the robots and computing facilities at LAAS. Thanks to Natacha, Camille and Helene for always helping me with administrative things at the lab.

I'm very grateful to my friends from LAAS: Ali, Lavi, Assia, Aamir, Umer, Amit, Red, Layale, Cyril, Gil, Bob, David, Diego, Matthieu, Ibrahim, Alhayat, Ela, Antonio, Hayat, Wassima, Xavier, Hung, Bach Van, Jean-Marie, Arnaud, Aurelien, Wassim, and Sakku for their help, care and more importantly the motivation and encouragement throughout my stay in Toulouse. I'm also very grateful to my friends Sofie, Matthieu, Solene, Merle, Bushra, Celine, Emma, Humaira, Umer, Saqlain, Saif, Tauseef, Kaleem, Nadeem, Onder, Gulfam, Sara, Tusawar, Hussnain, Saima, Ayesha, Ousama, Anne, Julie, Owais, Rosanne, Saad, and Tameez for their care, help, encouragement and the tasty dinners.

Finally a special thanks to my parents, my brothers and sister, my cousins, uncles and aunts for always being there for me, for their support, care, encouragement, advice, prayers and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous navigation in a mobile robotics context refers to the ability of a robot to move from one position to another desired position without the intervention of a human operator. Autonomous navigation ranges from simple tasks such as moving a few meters in a straight line in a known environment, to complicated tasks such as navigating autonomously in an a priori unknown environment for extended periods of time. A very basic functionality for a robot in order to perform autonomous navigation is that it should be able to perceive its surroundings and to localize itself in its environment, in order to navigate in a safe and desirable manner. Robots use different kinds of sensors in order to perceive their surroundings such as cameras, sonars, radars, and lidars. The introduction of 3D lidar sensors (such as the Velodyne HDL-64E S2 (90)) in robotics has revolutionized the way robots perceive their surroundings, as they provide rich 3D data at a fast rate. The main contribution of this thesis is the exploitation of such novel sensors in autonomous navigation. This includes a technique to calibrate them and a technique for efficient qualitative localization of autonomous robots using their data.

1.1 Context

1.1.1 Environment perception for outdoor mobile robots

An autonomous mobile robot necessarily has to perceive reliable and sufficient information on its surrounding in order to perform any navigation task. The "autonomous" nature of a robot makes it obligatory for the robot to rely entirely on its sensors in order to operate. This makes reliable environment perception crucial in autonomous robotics in comparison to manually controlled robots. For instance a very basic function for which reliable perception information is required during autonomous navigation is obstacle avoidance: without being able to reliably detect obstacles, a robot cannot avoid them.

Often autonomous mobile robots do not have a priori knowledge about the environment they are supposed to operate in. Therefore they have to create maps or models of the environment as they move around in the environment. Robots use information on their surroundings acquired by different sensors, to build such models of their environment. Building such models in indoor environments is usually simpler than in outdoor environments. Indoor environments are usually structured, having walls doors, corridors etc, and mainly consist of flat grounds. Therefore while building an environment model indoors, a robot might take the above mentioned characteristics of indoor environments for granted. Outdoor environments on the other hand are usually unstructured or semi-structured at best – "semi-structured" refers to the urban outdoor environments that contain structured objects like buildings as well as unstructured objects like trees etc. Moreover outdoor terrains are not systematically flat, and therefore a robot must take the terrain elevation into account while building an environment model. This makes outdoor environment modeling for autonomous navigation more challenging than the indoor environment modeling and an autonomous robot operating outdoors has to acquire rich perception information form its surroundings to be able to build an environment model.

In order to build spatially coherent environment models, a robot must also be able to localize itself correctly. While it might be possible for an autonomous robot to use localization aids like GPS (Global Positioning System) sensors, it is not always possible or desirable to let an autonomous robot rely on such sensor for localization: the precision of their position information is not good enough to build detailed environment models from the acquired data, and multipath errors caused by buildings and other structures yield erroneous position estimates. Therefore autonomous robots must rely on the information perceived on the environment to localize themselves without any external aid. A significant amount of work has been done on simultaneous localization and mapping (SLAM) for autonomous robots in the last couple of decades. SLAM refers to the autonomous navigation framework where a robot, as it starts to move in an a priori unknown environment, observes its surroundings and starts to build a map of its environment along with keeping an estimate of its location (and orientation) without the aid of localization sensors like GPS. In case a robot is a priori equipped with a model of the environment in which it has to operate, the robot uses perception information from the environment in order to localize itself in the environment by comparing the perceived data with the given model of the environment (the so called "map based localization" problem).

If an autonomous robot is capable of building a spatially coherent environment model (in case it is not available a priori) and localizing itself with respect to that model using the perception information, then it can autonomously navigate in that environment, *i.e.* plan the best (e.g. the shortest) path to reach a goal position and execute the planned path.

In a nutshell, it is imperative for a mobile robot to be able to acquire reliable and sufficient perception information from its environment in order navigate autonomously.

1.1.2 Sensors

Robots use different types of sensors in order to perceive their surroundings. This includes passive sensors such as cameras, and active sensors such as sonar (SOund Navigation And Ranging), radar (RAdio Detection And Ranging), and lidar (LIght Detection And Ranging). These sensors are listed below along with their basic characteristics.

Cameras. Cameras/vision sensors have extensively been used in robotics. Indeed They provide rich information about the environment, are low-cost, light and compact, are easily available and have low power consumption. All these qualities have lead to their extensive use in robotics.

Cameras are used in different forms including stereo-pairs, multiple camera rigs, as catadioptric sensors for panoramic vision and even as catadioptric sensors for stereo-panoramic vision e.g. (37) (*cf.* figure 1.2). A single cameras is a bearing only sensor *i.e.* it only gives directional information about features in the scene but when used as stereo-pairs or multi-camera rigs, the depth information can be also be extracted from the scene using triangulation. Even though vision sensors can be employed to extract depth information, the accuracy of the depth information acquired through

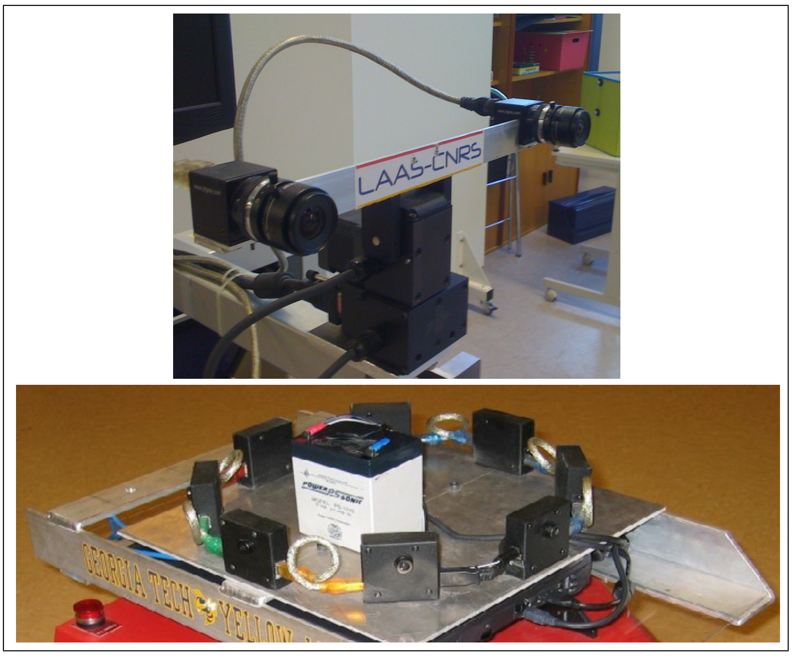

Figure 1.1: A stereo camera pair (top). An eight camera rig from (33) (bottom).

triangulation decreases rapidly with the depth of features in the scene. This makes vision sensors suitable for indoor navigation where the environments are structured and thus abundant in distinctive features and the observed distances are short.

Another limitation of cameras is that as they are passive sensors, *i.e.* they do not emit any kind of energy but detect the light present in the environment, and are thus dependent on the illumination conditions in the environment. This in turn decreases their robustness in robotic applications. For instance a vision based obstacle avoidance system that works perfectly fine under normal lighting conditions will fail in dark or even under very high illumination.

Sonars. Sonar sensors emit pulses of sound and listen for echos to detect and compute the range of detected objects. Sonars are slower than most other types of sensors used in robotics for perception purposes owing to the slower speed of sound in comparison to light or radio waves. Nevertheless sonar sensors have been used in robotics because they are simple to use and are relatively low-cost. Figure 1.3 (left) shows Millibot robot from (59) fitted with a sonar sensor. However sonars have a large angular ambiguity, and are not anymore used for the purpose of environment modelling and localisation –

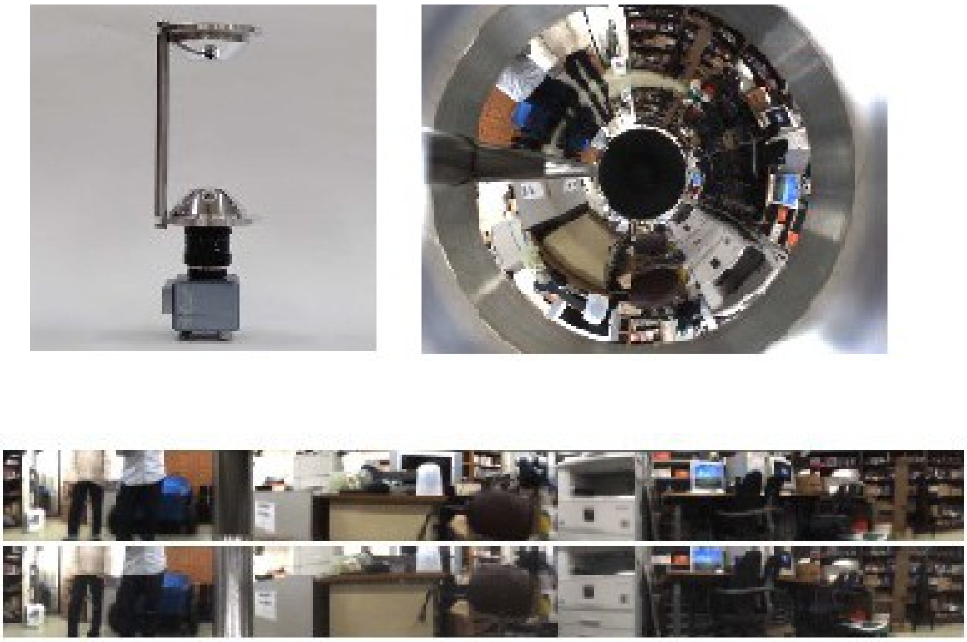

Figure 1.2: A single-camera-double-mirror catadioptric stereo sensor from (37): the sensor (top-left), an acquired image (top-right) and the corresponding two unwrapped panoramic images (bottom).

especially for outdoor robots, where their small range hinder their use.

Radars. Radars work on a similar principle as sonar sensors but use radio waves instead of sound pulses in order to determine the position of objects present in the scene. Radars have not been used as extensively in robotics as other active sensors such as sonars and lidars. (87) and (70) present the use of custom built radar sensors for environment perception in outdoor robotics.

Lidars Similar in principle to sonars and radars, lidar are active sensors that emit infrared light pulses and measures the time its takes for the pulse to be reflected from an object and return to the sensor in order to measure the depth of the object. Lidars sensors are attractive for environment perception in robotics because unlike cameras they readily provide depth information about the environment. Moreover the depth information provided by lidar sensors is more accurate than that provided by stereo-camera pairs. The conventional lidar sensors used in robotics consist of a single laser beam that employs a rotational mechanism to scan the environment in 2D. These lidar

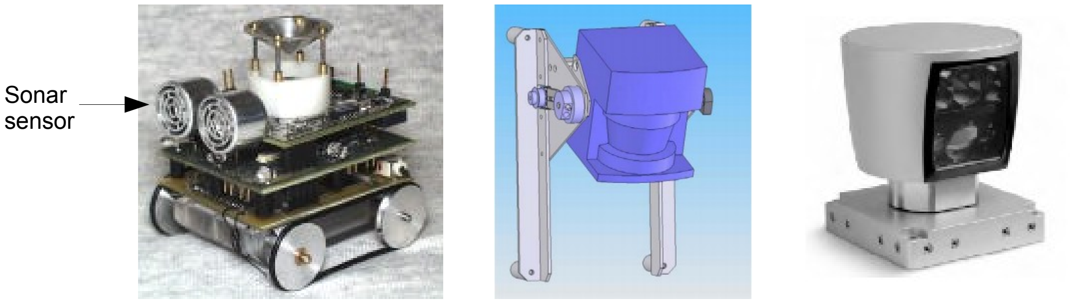

Figure 1.3: Millibot robot from (59) fitted with a sonar sensor (left), a tilt unit with a 2D SICK lidar from (94) (middle), and the multi-beam lidar sensor Velodyne HDL-64E S2 (right).

sensors are sometimes fitted on a tilt unit in order to acquire depth data from the environment in a solid angle (*cf.* figure 1.3 (middle) – such devices are often called “3D scanners”).

Recently, the introduction of multi-beam lidar devices in robotics has enabled the robots to acquire 3D depth information about their environments at a very fast rate. Velodyne HDL-64E S2 is one such example (*cf.* figure 1.3 (right)). The device consists of 64 laser beams located in a spinning head that can rotate at the speeds of 5-15Hz. The device has a 360°x27° field of view and it provides depth data at a rate of 1.33 million points per seconds. Figure 1.4 shows a sample scan acquired from the Velodyne lidar device along with an aerial view of the perceived environment. The introduction of such devices in robotics has revolutionized the way in which robots perceive their environments but it also raises several challenges. The first one is the intrinsic calibration of such sensors. The position of each laser beam with respect to the lidar device is defined by a set of geometrical parameters that must be known, and each laser beam has its own additional and/or proportional distance correction factor that arises from its internal electronics. Calibrating multiple parameters for multiple laser beams in such devices simultaneously is a challenging task. Another challenge in exploiting such sensors in robotics to their full potential is to develop techniques that can efficiently process and utilize the rich and large amount of data generated by such devices. For instance a great amount of work has been done on feature extraction from camera images but fast feature extraction techniques from rich and 3D data acquired but novel 3D lidar sensors are yet to be explored.

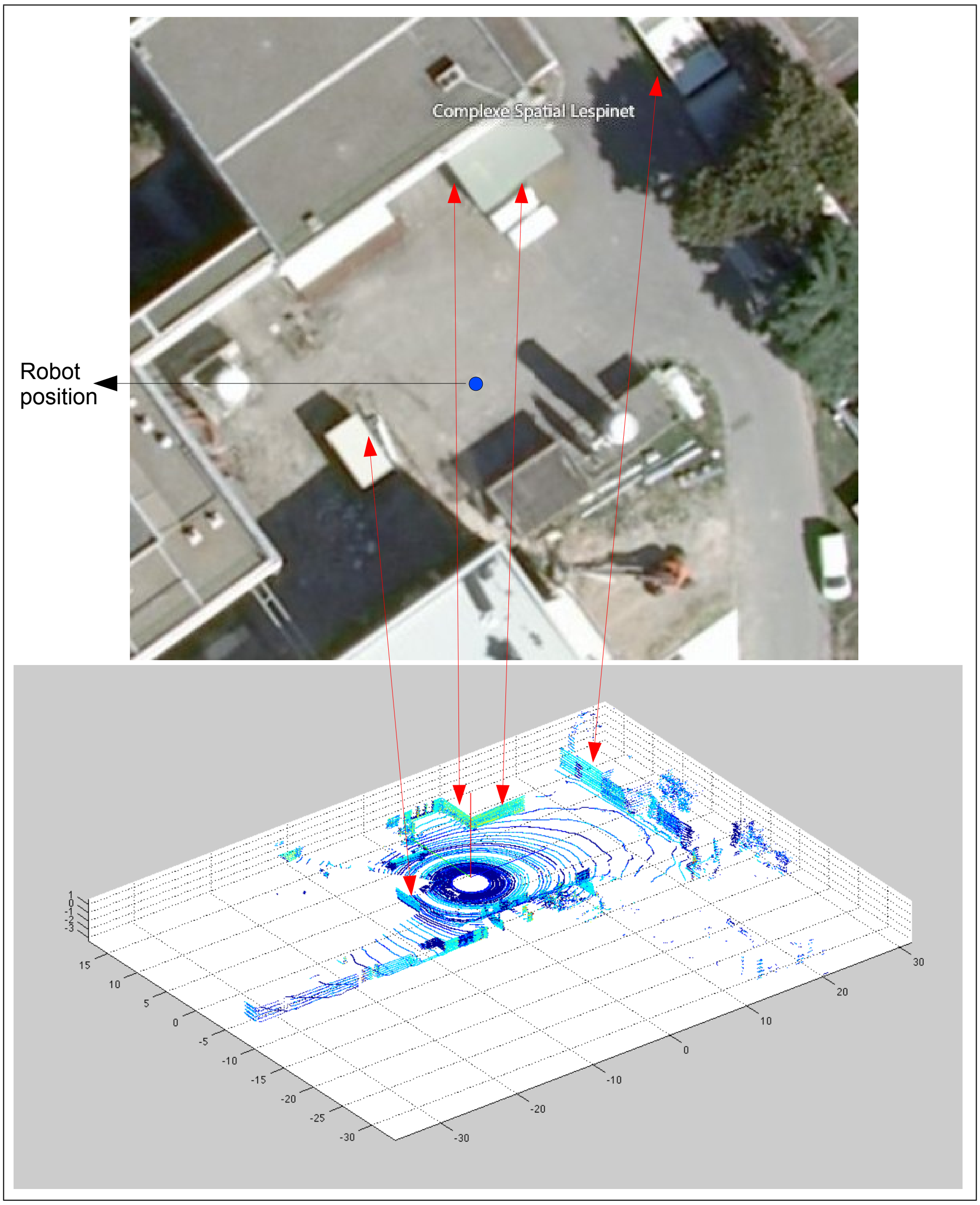

Figure 1.4: A sample scan from the Velodyne HDL-64E S2 lidar device in a semi-structured outdoor environment (units in meters).

1.2 Main contributions

Single beam 2D lidar sensors have long been used for environment perception in robotics but the fast multi-beam lidar sensors have only been introduced in robotics since 2006 when Velodyne introduced the HDL-64E lidar device. Since then such sensors have successfully been employed in robotics but their potential for autonomous robot navigation has not yet been explored as much as other perception sensors such as 2D lidars or vision sensors. This thesis focuses on the exploitation of 3D lidar sensors for autonomous outdoor navigation.

The contributions of this thesis are twofold. The first contribution is a method for intrinsic calibration of multi-beam lidar devices. The calibration of any sensor is of paramount importance for its successful exploitation for any task. In environment perception in robotics for instance, a well calibrated sensor can be exploited to its full potential during autonomous navigation whereas a badly calibrated sensor will provide imprecise perception information which might lead to the a mission failure. Intrinsic calibration of multi-beam lidar devices consists of estimating the geometrical parameters that define the laser beam position and orientation inside the lidar device as well as estimating the distance correction factors for individual beams. This thesis presents a method for calibrating the geometrical parameters defining the laser beam orientation inside the multi-beam lidar devices as well as the estimation of distance correction factor for the individual laser beams. The method is based on the optimization of calibration parameters by comparison the acquired 3D data to the ground truth model of the calibration environment.

The second contribution of the thesis is a method for the extraction of small-sized global signatures from 3D lidar data for view-based localization. View-based localization is performed either by extracting global signatures from the environment or by extracting local features and representing the robots surrounding as a set of local features. For instance an image captured by a robot can be encoded as a histogram of color information to serve as a global signature, or a corner or edge detector can be applied on the acquired image in order to extract local features from the surroundings of a robot. This thesis presents a method for extraction of small-sized global signatures from Velodyne lidar data. The global signatures are based on the local surface normal information that can efficiently be extracted from the captured point clouds by exploiting the laser beam arrangement inside the Velodyne lidar device. The thesis then presents a method for qualitative view-based localization using the defined global signatures. “Qualitative” in the context of localization refers to the fact that the localization information provided by the technique is not with respect to a metric map but with reference to the places already visited by the robot.

1.3 Thesis structure

The thesis is divided into two parts, each consisting of two chapters.

- **Part I** of the thesis addresses the calibration of multi- beam lidar devices and is structured as follows:
    - **Chapter 2** presents an overview of the use of lidar devices in robotics. The chapter also presents the Velodyne HDL-64E S2 multi-beam lidar device and gives an account of the usage of this and similar devices in robotics.
    - **Chapter 3** presents a method for the intrinsic calibration of multi-beam lidar devices. The chapter also presents the calibration results for the Velodyne HDL-64E S2 lidar device.
- **Part II** of the thesis addresses the exploitation of multi- beam lidar devices in robot localization and is structured as follows:
    - **Chapter 4** describes the importance of localization for autonomous robots and presents an overview of various solutions to the localization problem. The chapter also presents the concept of view-based localization in robotics.
    - **Chapter 5** presents a technique for the extraction of local-surface-normal-based global signatures from 3D lidar data. The chapter details how local surface normal information can efficiently be extracted from 3D lidar data by exploiting the arrangement of laser beams inside the Velodyne lidar device. The chapter also presents a technique for qualitative view-based localization of an autonomous robot using the aforementioned global signatures.

Finally **Chapter 6** concludes the manuscript and discusses possible extensions to the presented work.

Part I

Lidar Calibration

Chapter 2

Lidar in Robotics

Lidar sensors have been extensively used in robotics. They are active sensor that emit and detect (usually infrared) light to estimate the distance of objects. This chapter presents the principle of lidar operation and a short description of some lidar devices used in robotics. The chapter also details the existing work on lidar calibration and describes the functioning and applications the Velodyne HDL 64E S2 lidar device, which we have been using throughout the thesis work.

2.1 Lidar devices used in robotics

The principle of operation of lidar sensors and some lidar devices used in robotics are presented below.

2.1.1 Principle of operation

Lidar stands for ”LIght Detection And Ranging”, and work on the principle of time of flight measurement. A laser diode emits an infrared (typically 905nm wavelength) laser pulse which is collimated by a transmitter lens. The emitted laser beam hits a target and a part of the reflected light hits a photodiode after passing through a receiver lens. A precise clock is used to measure the time between transmitted and received signal which in turn is used to compute the target distance from the device. The intensity of the received signal is also used to measure target characteristics such as reflectivity. This process is shown figure 2.1.

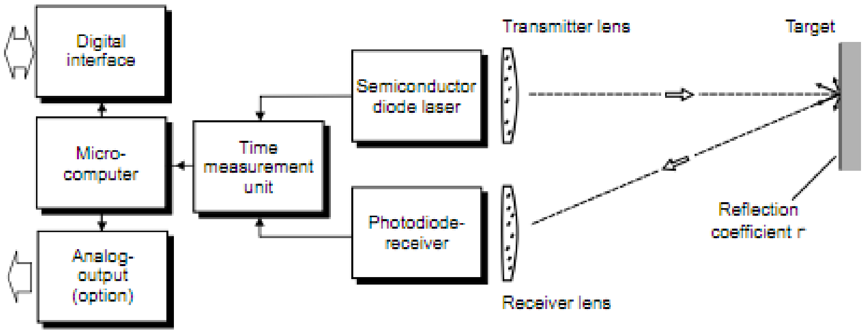

Figure 2.1: Lidar principle of operation (from (67)).

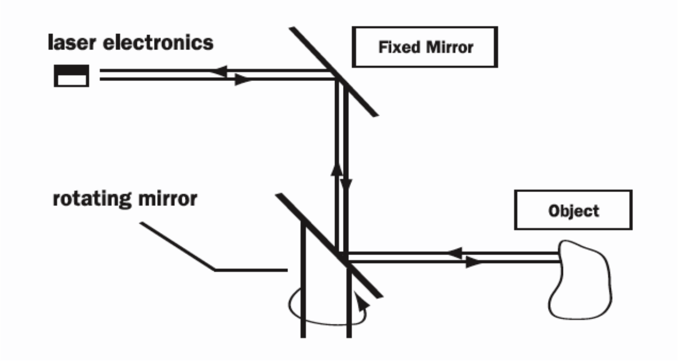

Figure 2.2: Single laser with rotating mirror mechanism to cover an angular field of view (from (89)).

The above principle can be extended in different ways to form more sophisticated lidar sensors. This includes using multiple transmitter/receiver pairs and using rotation mechanisms to increase the sensor field of view. Figure 2.2 shows a rotating mirror mechanism typically employed with a single laser beam to cover a wide angular field of view.

2.1.2 Lidar devices used in robotics

2.1.2.1 SICK LMS2xx

This is perhaps the most commonly used class of lidar devices used in robotics so far. These lidar devices use a single laser beam and a rotating mirror to cover an angular field of view of up to 180°. The LMS200 variant is suited for indoor application. The device can measure up to 80m range for high reflectivity targets but only up to 10m for targets

Figure 2.3: SICK LMS200 (left) and LMS291 (right)

Figure 2.4: The Hokuyo UTM-30LX lidar

with 10% reflectivity. The LMS291 variant is suited for outdoor applications. It also has 80m operational range but an enhanced 30m range for objects with 10% reflectivity. The device also has a fog correction feature that enables it to work under harsh weather conditions. Both types of devices can provide (selectable) angular resolutions of 0.25°, 0.5° or 1° and 10mm measurement resolution. The LMS200 has a typical measurement error of ±15mm and for the LMS291 this value is ±35mm (77). Both devices weigh around 4.5 kg and are shown figure 2.3.

2.1.2.2 Hokuyo UTM-30LX

This is another lidar device for robotic applications with a range of 30m and 270° angular field of view. It has an angular resolution of 0.25° and ±50mm measurement accuracy (29). The device weighs only 370g and is thus also suitable for small robots (figure 2.4).

Figure 2.5: The Ibeo LUX lidar

2.1.2.3 Ibeo LUX

Another lidar device conceived for applications in the automotive industry is the Ibeo LUX ((23), figure 2.5). It has an average operational range of 200m, and 50m for targets with 10% reflectivity. It has a vertical field of view of 3.2° and can scan the environment with two or four layers with horizontal fields of view of 110° and 80° respectively. It has a horizontal angular resolution of 0.125° and distance measurement resolution and accuracy of 4cm and 10cm respectively.

2.1.2.4 Velodyne HDL-xxE

These are very novel and promising lidar devices that scan the environment in 3D at very high data rates. The device originated during DARPA Grand Challenge 2005 when the Velodyne *TeamDAD* used atop their autonomous vehicle a prototype of what would become the Velodyne HDL-64E in future. The device had 64 lasers, and it rotated at 10Hz and scanned the environment with a 360°x20° field of view (27). The device was 30cm in diameter and weighed around 45.3 kg (88). Velodyne then marketed a more compact version of the device in 2006 as the HDL-64E and as HDL-64E S2 in 2008 that weighs around 13 kg (90). The device also has 64 laser/detector pairs and has a 360°x27° field of view. It provides range measurements at a rate of 1.33 million points per second and can rotate at a user selectable speed between 5 to 15Hz. A detailed description of the device is presented in section 2.3. In 2010 Velodyne introduced another smaller lidar device, the HDL-32E. It has 32 laser/detector pairs and has a 360°x40° field of view (91) and provides range measurements at a rate of 700,000 points per second. The device weighs less than 2 kg. Figure 2.6 shows the Velodyne lidar prototype, the HDL-64E S2 and the HDL-32E models.

Figure 2.6: The Velodyne lidar prototype (left), HDL-64E S2 (middle) and HDL-32E (right).

Figure 2.7: MESA SwissRanger SR4000

2.1.2.5 Flash lidar devices

These devices are essentially 3D cameras that capture range and intensity information of the scene lying in their field of view in a single shot, unlike conventional 3D lidar devices that do it point by point. These devices emit a laser pulse in a solid angle and the reflected light is detected by a pixel array of photodiodes. As these devices have no mechanically moving parts, they have several advantages over conventional 3D lidars such as their light weight and low power consumption. On the other hand these devices typically have shorter operating ranges and fields of view compared to the conventional lidar devices. One such device is the SwissRanger SR4000 from MESA Imaging ((54), figure 2.7). It captures 3D images with a QCIF (176x144 pixels) resolution. It comes in two measurement range versions of up to 5m and 10m and two field of view versions of 43.6°x34.6° and 69°x56°.

Figure 2.8: Riegl LPM-321 (left) and Konica Minolta Range5 (right)

2.1.2.6 Industrial and survey lidars

Long range lidar devices are used for surveying and mining applications. One such device is the Riegl LPM-321 which has a range of up to 6000m and a field of view of 360°x150° ((68), figure 2.8). It scans the environment at a rate of 1000 points per second. Such devices are relatively slow, and employ high-precision and fragile pan-tilt mechanisms in order to scan the environment: they are thus not suitable for exteroceptive sensing in mobile robotics. These mechanisms are normally designed for stationary operation of the device and thus are not rugged enough to be employed onboard outdoor robots running around on rough terrains. Similarly some lidar devices are conceived for 3D scanning of objects such as castings and archaeological items and provide very accurate range measurements. These devices also have slow scan rates and short measurement ranges which makes them infeasible for applications in mobile robotics. One such device is the Konica Minolta Range5 which has a measurement accuracy of $\pm 80\mu$m and a range of 450mm to 800mm ((38), figure 2.8).

2.2 Lidar Calibration

Calibration of exteroceptive sensors such as cameras and lidars consists of estimating its intrinsic and extrinsic parameters. The intrinsic parameters represent the internal geometric properties of a sensor, whereas the extrinsic parameters represent the position and orientation of the sensor in a robot-fixed frame and thus allow the transformation of individual sensor readings to the robot-coordinate frame. For a camera for example, intrinsic parameters model its internal geometry and lens characteristics, whereas extrinsic parameters model its position and orientation in a world-fixed frame (73). The calibration is normally a two-step process *i.e.* modeling of the sensor parameters and their estimation using direct or iterative methods (73). Usually in camera calibration the camera is modeled as a pin-hole. This means that the camera aperture is a pin-hole and thus all the light rays intersect at one point before being projected onto the image plane. Some more generic calibration techniques also exist where a camera is modeled as a collection of lines that do not intersect in one point (79). In this case camera calibration boils down to the determination of coordinates defining these lines in a common coordinate system ((73) present a comparison of different camera calibration methods). (7) is a widely used toolbox for the calibration of intrinsic and extrinsic calibration of cameras. Using the toolbox, several arbitrarily taken images of a planar checker-board lead to the determination of the intrinsic calibration parameters of the camera.

Just like cameras, in lidar calibration the intrinsic parameters represent the internal geometric properties of the device *e.g.* orientation of the emitted laser beam and the properties of the receptor photo-diode. Similarly the extrinsic parameters represent the position and orientation of the device with respect to the robot-fixed frame. Although a lot of work has been done on the calibration techniques for cameras, calibration techniques for laser and especially multi-laser systems have not been investigated to that extent. Some techniques for extrinsic and intrinsic calibration of single and multi-beam lidar devices are presented below.

(85) present a technique for extrinsic calibration of a Sick LMS-291 lidar mounted on a mobile robot. In principle the technique consists of observing an environment with known or partially known geometry and optimising the calibration parameters to minimise the difference between observed data and the known geometry of the environment. A simple test environment is constructed consisting of flat ground and a vertical pole covered with retro-reflective material shown figure 2.9. The retro-reflective material eases the distinction between scan data from the ground and the pole, leading to automatic segmentation of the two data. To quantify the similarity between scan data and real environment a cost function c is defined consisting of two parts accounting for the data from the pole and the ground. The cost for the pole data c_{pole} is the average squared perpendicular distance of scan points to the pole, where the pole is considered to be located at the position defined by the mean of all the points in the pole data.

Figure 2.9: Calibration environment (from (85))

Similarly the cost for the ground data c_{ground} is the average squared distance of the ground points (taking only $z-coordinates$ of ground points) to the ground plane where the ground plane is represented by mean z value of all the points in the ground data. The total cost c is the sum of cost values for the pole and ground data. The robot platform fitted with the lidar device moves in the environment, and scans it from multiple perfectly localized points of view. The scan data is then used for performing optimisation *i.e.* minimising the cost by adjusting the sensor extrinsic calibration parameters and thus finding their optimal values. Figure 2.10 shows the pole data scanned from multiple viewpoints before and after the calibration process (85).

The work from (85) is extended to incorporate multiple sensors including groups of homogeneous and heterogeneous sensors in (86). The authors suggest that while using multiple exteroceptive sensors on board a robot, if their extrinsic parameters are calibrated separately a systematic contradiction might still exist while fusing the data from different sensors and therefore the optimal way is to calibrate the sensors simultaneously. The simultaneous calibration of a group of four Sick lidars (homogeneous sensors) and one Sick lidar and a radar (heterogeneous sensors) are presented in (86).

(78) present a procedure for calibration of the additive and proportional distance correction factors for their Velodyne HDL-64E lidar device. The authors observed that their lidar device added an error of 2.6cm for each 1m distance being measured. The lidar device is calibrated by first calibrating a few lasers that have almost zero pitch

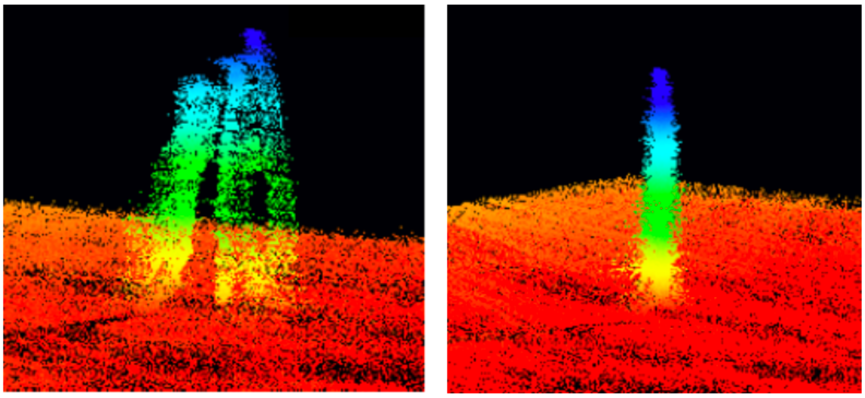

Figure 2.10: Pole in the calibration environment seen from multiple viewpoints before (left) and after (right) calibration (from (85)).

angles using indoor data sets with a vertical target at known distance from the device. Once this subset of lasers is calibrated, several data sets are taken by mounting the lidar device at 45° and scanning flat tall buildings. Using this data the two distance correction parameters are calibrated for the uncalibrated lasers considering the data from the previously calibrated lasers as the ground truth.

(6) mention the calibration of the distance correction parameters for a multi-laser scanner by comparing its distance readings to those from a Sick lidar but do not provide any details on the calibration procedure.

Similar "distance correction" calibration parameters have been reported for range imaging using time-of-flight cameras. (34) calibrate these parameters by making a look-up table for the operational range of device. (46) calibrate these parameters by fitting a B-spline to the measurement errors made by camera at different distances in the operational range.

(22) present a method for calibration of Velodyne HDL-64E S2. Six calibration parameters (rotation and vertical correction angles, horizontal and vertical offsets, and additive and proportional distance correction factors; described in detail in section 2.3.1) per laser beam for the device are calibrated using a planar feature based least square adjustment. The calibration data set consists of sixteen scans captured from two locations within a courtyard between four identical buildings. At each of the two locations four sets of data are collected by first mounting the lidar device horizontally and then by tilting it at an angle of 30°. The authors mention that four sets of data at

each mounting position consists of data taken at 0°, 90°, 180° and 270° azimuth but it is not clear whether this means subsets (e.g. segmented planar surfaces) of a 360° scan at these azimuth angles or four complete 360° scans by moving the lidar device through these angular positions (in which case these four scans would be identical since the lidar has a 360° field of view). Six unknown per scan location are also added to the total number of parameters to be estimated as the locations from which the data are taken are considered to be unknown a priori. The system is then constrained by holding the rotation and vertical correction angles, and the horizontal and vertical offsets for one laser beam, as well as the position and orientation of two scan locations constant. A unit-length constrain is further added to the system for the unknowns parameters of planes on which lidar points are conditioned. The least squares based adjustment is then performed to obtain the optimal values for calibration parameters. The authors report a three-fold improvement in RMSE for planar-misclosure in scan data.

(43) present a method for unsupervised calibration of multi-beam lidars. The technique is implemented for extrinsic and intrinsic (three intrinsic parameters *i.e.* rotation and vertical correction angles and additive distance correction factor) calibration of a Velodyne lidar. The technique is based on the assumption that points in space tend to lie on contiguous surfaces. A robotic vehicle fitted with the lidar device moves and scans data in an environment which is assumed to be static and containing 3D features and not just flat ground. All scan data are accumulated using the local pose information acquired from an GPS/IMU unit. The estimation of the calibration parameters is based on an energy function which penalises the points from a specific laser beam if they are far away from nearby points acquired by neighbouring beams. Using this energy function, first the extrinsic calibration parameters are estimated, then the three intrinsic calibration parameters are estimated for all the 64 lasers in three independent steps. Although the authors present a generalised method for calibration of multi-beam lidars such as the Velodyne, the results presented are only a qualitative comparison to the calibration data provided by manufacturer for the Velodyne device used in experimentation. The authors also present the calibration of intensity returns from each laser by deriving a Bayesian model of the beam response to surfaces of varying reflectivity.

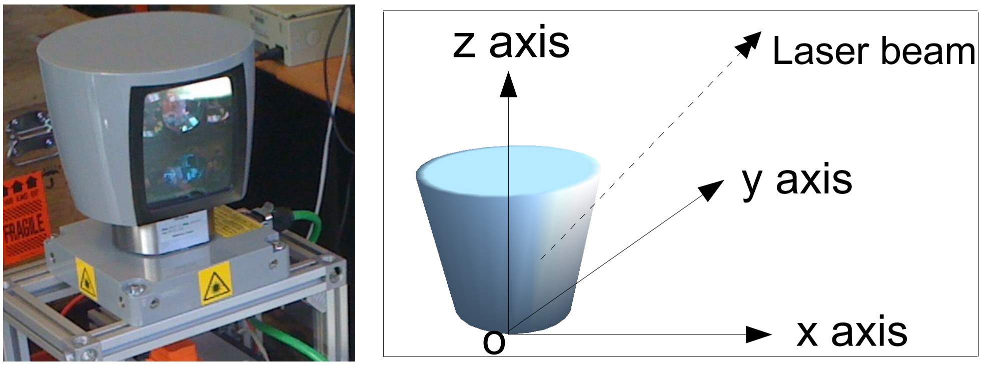

Figure 2.11: Velodyne HDL-64E S2 (left) and sensor-fixed frame (right)

2.3 The Velodyne HDL-64E S2

The lidar device Velodyne HDL-64E S2 scans the environment in 3D with a 360° azimuthal and 26.8° vertical field of view. The device has 64 lasers distributed vertically to cover 26.8° vertical field of view. The lasers are located in a spinning head that spins at rates of 5 to 15 Hz, to cover 360° azimuthal field of view. The device outputs data at a rate of 1.33 million points per second, the data rate being independent of the rotational speed of lidar head. Therefore at lower spinning rates, the lidar scans the environment with a higher spatial resolution and vice versa. An interesting point to note is that the device only scans the environment when spinning at sufficiently high speed. The lasers in the device are not fired when it is rotating at low speeds during device start-up. (71) observed that after start-up, it takes around 7 seconds before the device starts firing its laser beams and the time it takes for the device to attain stable operational speeds of 5, 10 and 15 Hz is 11.5, 18.5 and 32 seconds respectively. The device is class 1M eye safe and runs on a 12 V input drawing about 4 amps of current during normal operation.

2.3.1 Geometric model

64 lasers inside the lidar device are grouped in two blocks, with 32 lasers located in the upper block and remaining 32 in the lower block on the lidar spinning head. Lasers in each block are further divided into two groups of 16 lasers each, with one group located towards right and one towards left in each block on the lidar head. Figure 2.12 from (81) shows the device without its enclosure. The figure shows four

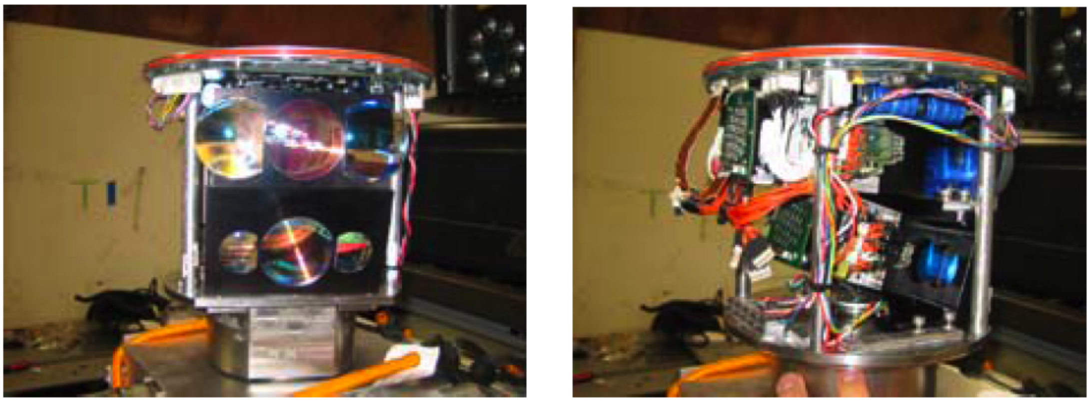

Figure 2.12: Velodyne lidar interior from (81): The figure shows four transmitter lenses corresponding to the four laser groups each. For both the upper and the lower block, a receiver lens is located between the transmitter lenses of right and left laser group of the corresponding block.

transmitter lenses (corresponding to the four laser groups of 16 lasers each) and two receiver lenses (corresponding to the upper and lower laser blocks) along with other electronic components inside the device.

Figure 2.13 shows how the 64 lasers of the lidar device are spatially arranged to cover a wide vertical field of view.

2.3.2 Sensor behaviour and characteristics

According to the Velodyne documentation (90), the max range for laser returns from low reflectivity objects is 50m. The documentation also states that all returns less than 3 feet (around 0.9 meters) must be discarded. In practice if objects are located around the device at very short distances (within a couple of feets), this results in false data points at long ranges (around 120 meters). Figure 2.14 shows this phenomenon in a data set that was taken in our lab. The blue data points present in the centre is the real data spanning an area around 20 m long, and the red data points are all false and actually do not exist in the scanned environment. Another characteristic of these false data points is that they have high intensity values. These false data points can easily be discarded as they lie far beyond the normal range of operation of the lidar device. (6) also mention the sporadic existence of long range outliers for their Velodyne lidar device which are perhaps similar long range phantom points.

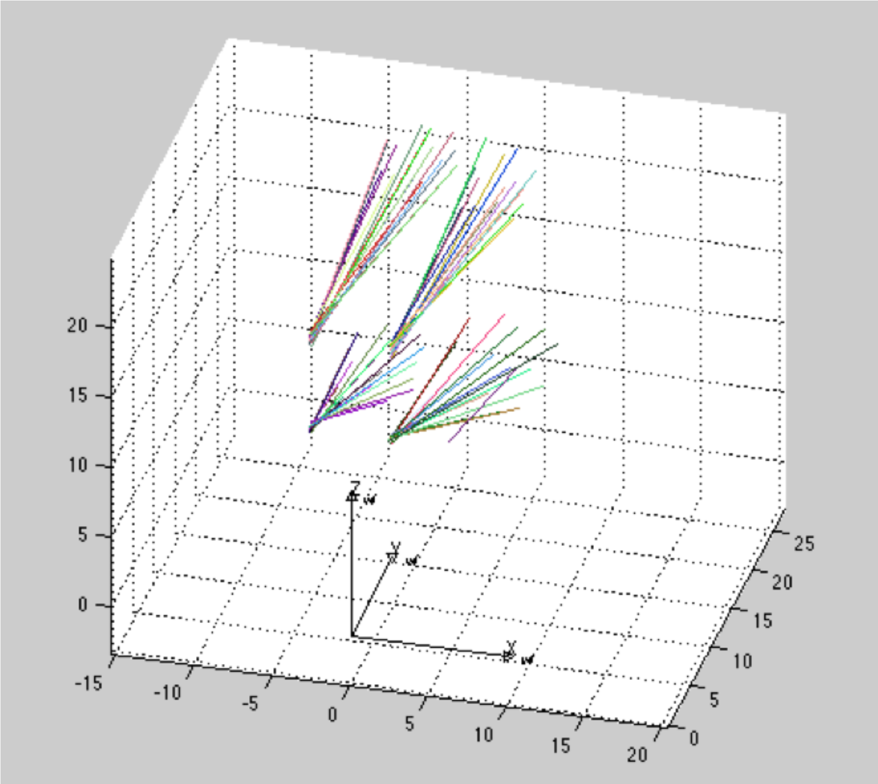

Figure 2.13: Velodyne HDL-64E S2: Four laser groups of 16 lasers each (units in cm).

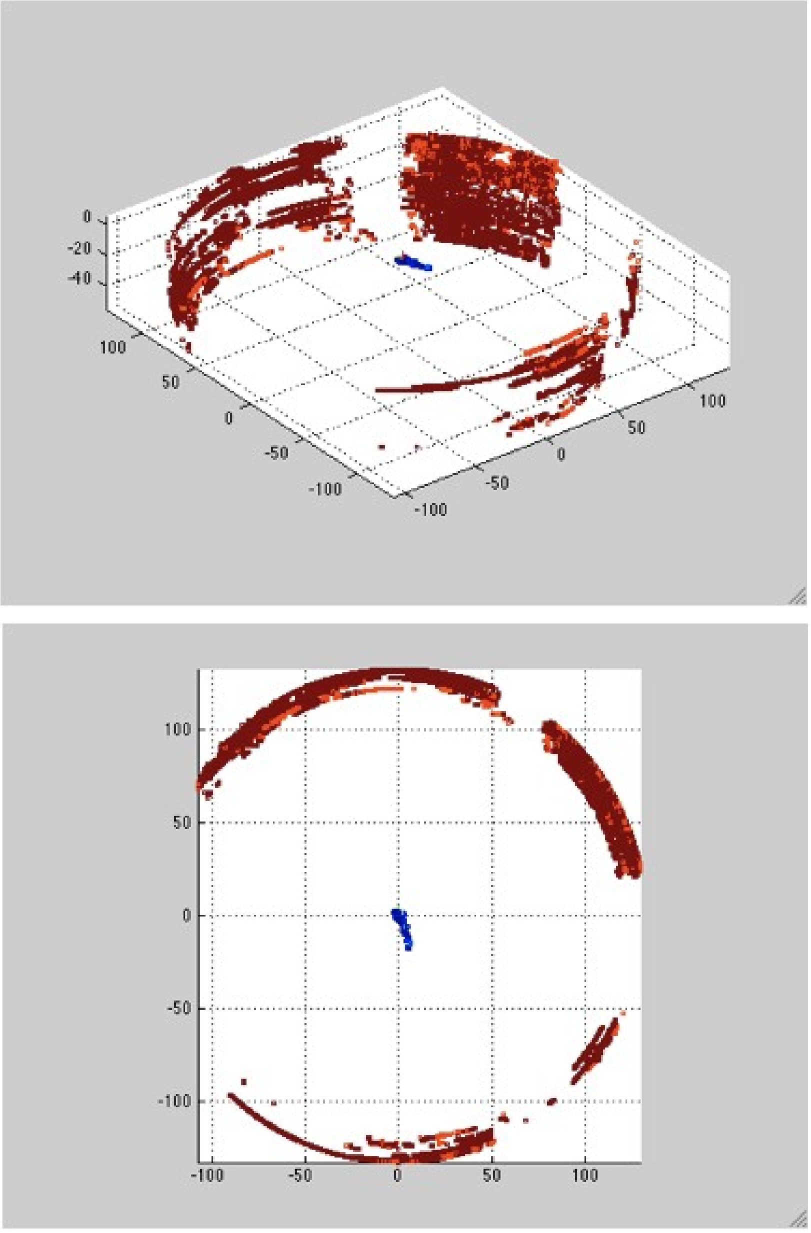

Figure 2.14: The real data (blue) and long range false data points (red), units in meters.

Table 2.1: Change in intensity returns with incidence angle

Incidence angle (degrees)	Average intensity value
0	45.6
20	42.1
40	39.4
60	33.6
70	30.1

The default calibration parameters provided by the manufacturer for each of the 64 lasers in the device are not very accurate. This is obvious in scan data if the scanned environment contains objects with regular geometry. Figure 2.15 shows two such examples. At top, the figure shows segmented view of a planar surface placed at 3m distance from the lidar device. Each horizontal line in the scan shows the data from a single laser, one can see that the data from all the lasers are not aligned to form a rectangular planar surface: instead the data from different lasers are horizontally misaligned, and the misalignment seems to be repeated between every two consecutive lasers. As the distance between the device and the plane increases, the misalignment first decreases until a distance of around 9m, and then increases but in the other direction between consecutive lasers (but this change in misalignment is not linearly proportional to distance). One possible explanation for this misalignment are the biases arising from unprecise calibration data.

Figure 2.15 (bottom) shows the top view of a corner in a room. Ideally, if the calibration parameters for all lasers were accurate and the sensor had lower noise, as seen from the top, the walls should have looked much thinner and corners much sharper.

For every distance measurement, the device also returns an intensity value. Theoretically the returned intensity value is a function of the emitted laser intensity, the laser incidence angle with the target and the target reflectivity and distance from the lidar device.

The intensity values returned by the Velodyne lidar device for a given object decrease as the incidence angle increases. Table 2.1 shows the change in intensity values with respect to incidence angle for a planar object placed at 4m distance from lidar.

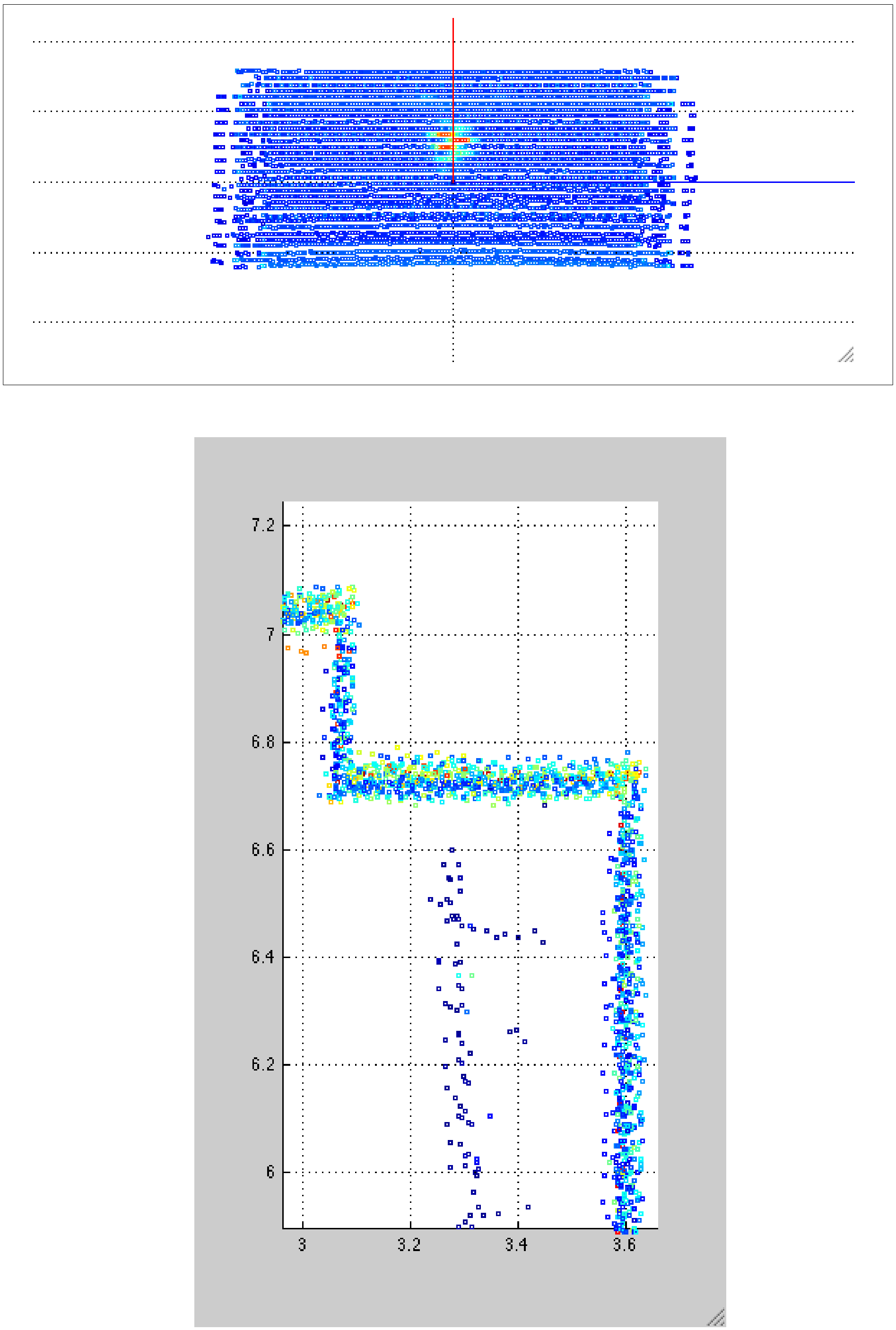

Figure 2.15: Segmented view of a rectangular surface (top) and top view of a corner in a room (bottom), units in meters.

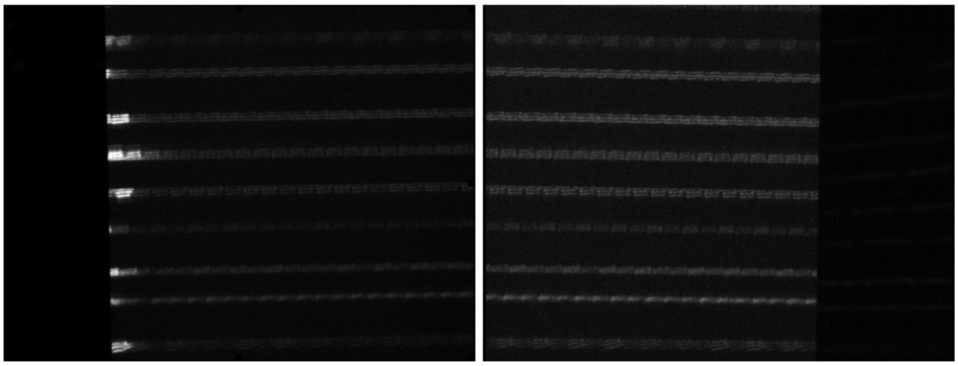

Figure 2.16: Phenomenon of intensity auto-adjustment (from (71))

At medium range in general (5m to 30m), the returned intensity values for a given target increase with as the target distance increases. This seems counter-intuitive if the intensity of the emitted laser beams is considered constant at all times. The device actually uses some automated gain control techniques to adjust the emitted intensity with respect to the distance observed in order to avoid saturations. This phenomenon of intensity adjustment was also observed by (71) for their Velodyne HLD-64E. Figure 2.16 shows the image of the left and right edges of a planar surface being hit by Velodyne laser beams. We can see that the intensity of emitted laser beams is higher when it started hitting the planar surface (left edge) but it is adjusted to a lower value right away. Table 2.2 shows the change in intensity value with respect to distance for a given target object.

When highly reflective objects are viewed by the device, the distance measurements returned by the lidar are not very accurate. These objects can show up at distances far from their original positions or as other phantom points. (71) also observed some phantom points because of highly reflective material attached to a planar surface being scanned. The phantom points appear below ground at a position which is in fact mirror of their actual location. The phenomenon is shown figure 2.17.

2.4 Velodyne lidar applications in robotics

As mentioned earlier the Velodyne HDL-64E S2 originated during DARPA Grand Challenge 2005 when the Velodyne *TeamDAD* developed a multi-beam lidar prototype for their autonomous vehicle. The prototype device was conceived and designed to ease

Table 2.2: Change in intensity returns with distance

Distance (m)	Average intensity value
2	49
4	46
6	61
9	107
12	114
15	150
20	168
25	203
29	238

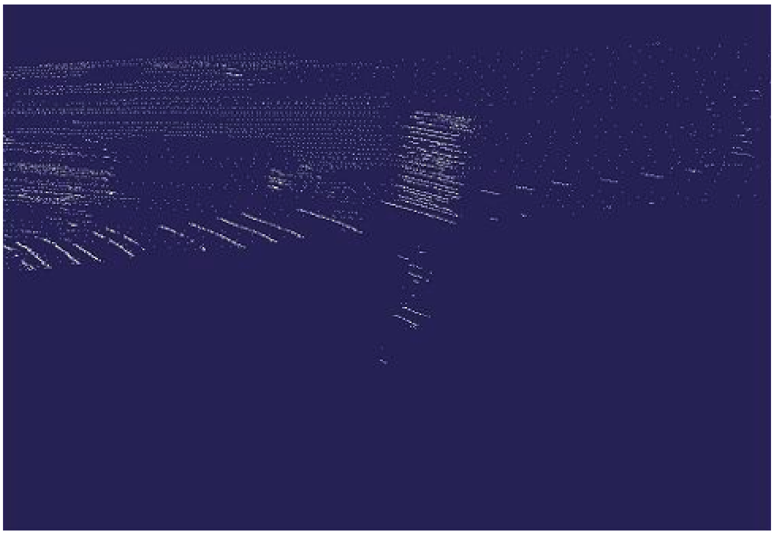

Figure 2.17: Phantom points because of very high reflectivity targets (from (71))

robust terrain mapping and obstacle detection and avoidance compared to that done using conventional single beam lidar and vision systems. The device had 64 laser, rotated at 10Hz and scanned the environment with a 360°x 20° field of view (27). The device had an INS sensor mounted on it to provide precise roll and pitch for the unit enabling any required corrections in lidar data used for autonomous navigation. Sunlight was filtered out before analysing the laser returns making the device robust to be used under all lighting conditions and the rotational mechanism of the device helped spinning off rain water and dust from the device, thus helping a clearer visibility under harsh weather conditions. Moreover the device rejected early laser returns in order to get more accurate range measurements under rain or fog. Range returns from the device were used to create a $x, y, height - value$ terrain map enabling the vehicle to determine size and distance of objects in view. No object classification was done and all the visible objects were considered solid. The terrain map was in turn used for obstacle avoidance and optimal road surface detection in accordance with the GPS way points provided to the vehicle.

This innovative lidar concept and its successful usage in perception for autonomous vehicle lead Velodyne to design and market a more compact version of the device *i.e.* the HDL-64E in 2006 which was used by many contestants of the 2007 DARPA Urban Challenge. Since then the device has increasingly been used in outdoor robotics in general and especially for full-size autonomous vehicles in urban areas. The device has also been used in maritime scenarios. An overview of the device usage in robotics and some other applications is presented below.

During the 2007 DARPA Urban Challenge six teams successfully completed the course and five of these used a HDL-64E as one of its exteroceptive sensors. Event winners were the Tartan racing team with their vehicle called “Boss” (19) shown figure 2.18. The not only employed the HDL-64E but also Sick LSM-291, Continental ISF 172 and Ibeo Alasca lidar devices along with other Ma/Com and Continental ARS 300 radars and MobilEye vision system. Along with other sensors the data from HDL-64E was used for tasks including (i) determining how safe is it to crossing and merging at intersections, (ii) detection and localisation of other vehicles and keeping a safe distance to them, (iii) estimation of road shape and lane locations and (iv) detection of static obstacles. Moving obstacle fusion architecture consisted of three layers. A “sensor” layer processed the data from each sensor independently and associated the

Figure 2.18: Tartan racing team vehicle “Boss” for the 2007 DARPA Urban Challenge (from (19))

measurements to the objects currently being tracked. A“fusion” layer then took the associated and unassociated measurements from the “sensor” layer and applied them to the global list of tracked objects. Then a “situation assessment” layer estimated the intention of the tracked object. The architecture is shown figure 2.19. States of moving obstacles were predicted and updated using Extended Kalman Filtering. Static obstacle detection was done using cost map representing the traversability of the terrain around the vehicle and the road-lane detection is done owing to the fact that the lidar intensity returns from painted road lines is very high.

Stanford racing team placed second in the 2007 DARPA Urban Challenge with their vehicle “Junior” (20) shown figure 2.20. Along with an HDL-64E the vehicle also used two Sick and two Ibeo Alasca XT lidars, and five Bosch long range radars. HDL-64E atop Junior was the primary obstacle detection sensor but had a blind spot behind the vehicle where Sick and Ibeo lidars supplemented the HDL-64E data. Overhanging objects like trees were filtered out by comparing their height with a ground model. Sensor measurements were cached into local maps to cope with the blind spots in the sensor fields of view. The maps were updated using a Bayesian framework for evidence accumulation. Such a map for a parking lot is shown figure 2.21. Moving object detection and tracking was done using a synthetic 2D scan of the environment which was synthesised using data from lidar sensors. Synthetic scans allowed for efficient

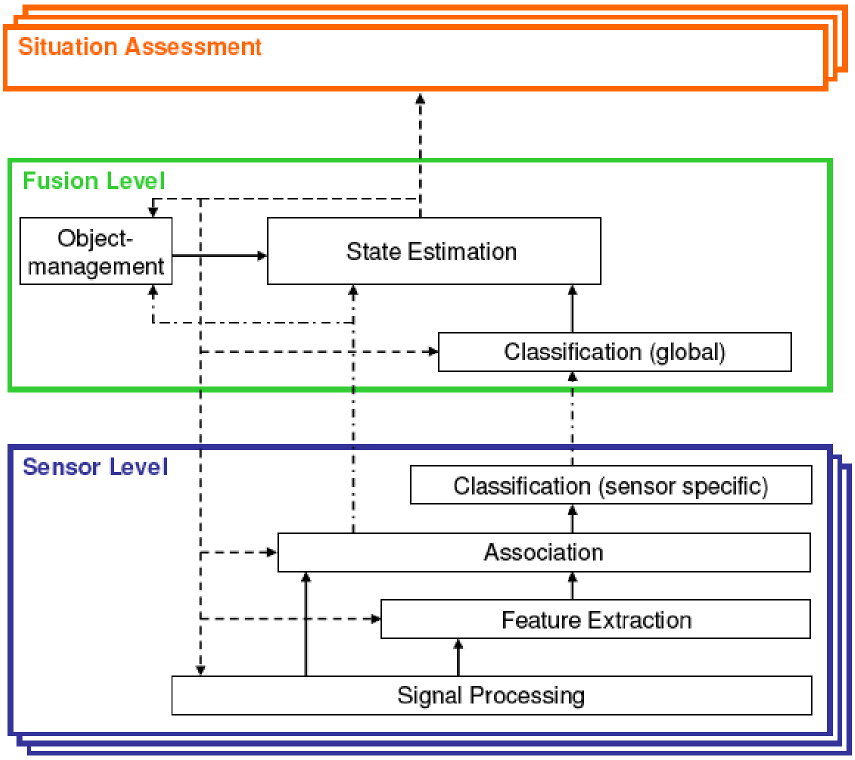

Figure 2.19: Three-layer moving obstacle fusion architecture of the vehicle “Boss” (from (19)).

Figure 2.20: Stanford racing team vehicle “Junior” for the 2007 DARPA Urban Challenge (from (81))

computation owing to their compactness compared to raw lidar data and also provided a single methodology for using data from Velodyne, Ibeo or Sick lidars or any of their combinations. Details on the Junior vehicle detection and tracking algorithms can be found in (64) and (65). (65) also describes how black objects (that are known to provide very small or no laser returns) can be detected and tracked using HDL-64E data. The logic used is that the absence of laser returns raises the presence of black objects, but it only works within a range of about 25-30m for HDL-64E data. The process is shown figure 2.22.

Similar accounts of Velodyne HDL-64E usage in the 2007 DARPA Urban Challenge by the Ben Franklin racing team, Austin Robot Technology, The Golem group LLC and team AnnieWay are presented in (6), (78), (48) and (35) respectively.

Research on autonomous vehicles has continued since the 2007 DARPA Urban Challenge and the primary perception sensor remains the Velodyne HDL-64E and its later version HDL-64E S2. One example is the work going on at Autonomous Labs located in Berlin, Germany. The enterprise is developing autonomous cars to be used as driver-less taxis (5). Another example is Google autonomous vehicles project (25). The motivation behind the development of autonomous vehicles is not only limited to the ease of use but also safer and reliable driving as autonomous driving systems are less prone to commit driving errors and thus reducing road accidents.

The Velodyne lidar can also be used in maritime applications such as obstacle

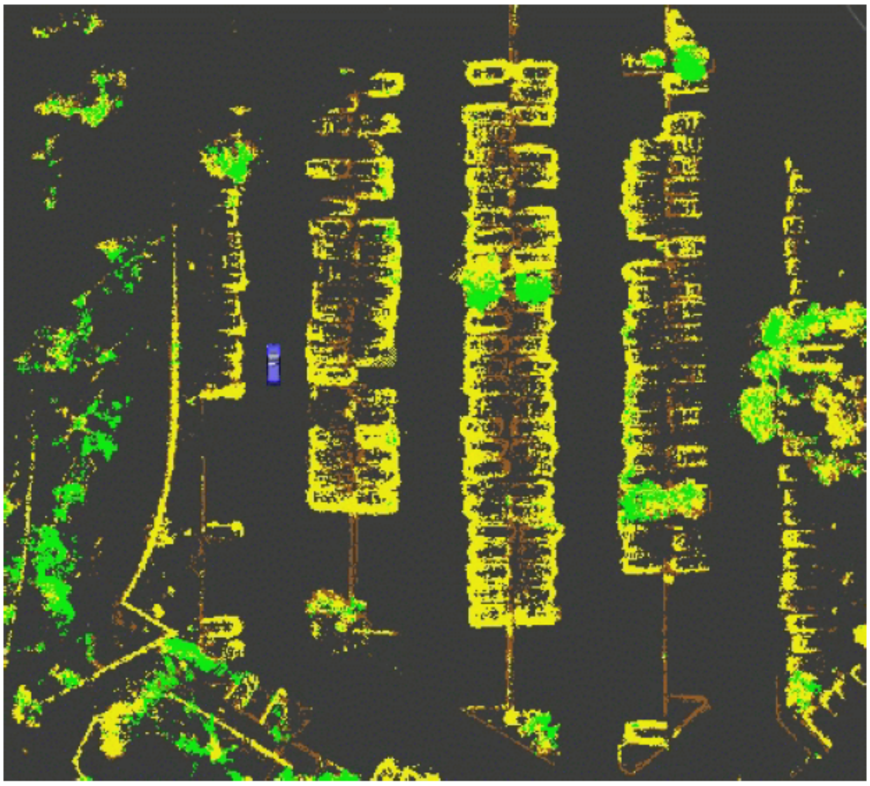

Figure 2.21: Junior static obstacle map showing a parking lot. Yellow colour represents tall obstacles, brown represents curbs and green represents overhanging objects (from (20)).

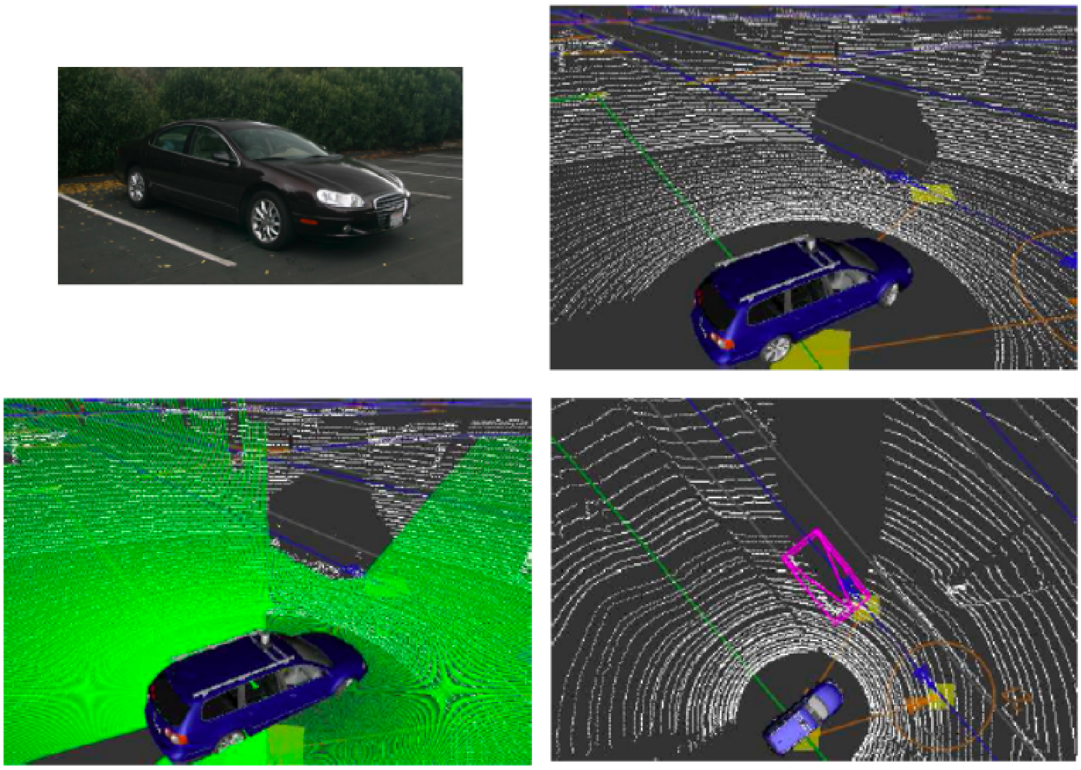

Figure 2.22: Tracking of black objects: a black vehicle (top-left), very few laser returns (top-right), virtual scan after black object detection (bottom-left), tracking of detected vehicle (bottom-right) (from (81)).

Figure 2.23: Maritime data acquisition using HDL-64E (from (28))

detection and avoidance for unmanned surface vehicles. (28) present the performance evaluation of HDL-64E for such maritime applications. The authors report that boats, kayaks and buoys can be detected using Velodyne data but it is not possible to detect fully submerged rocks in the sea because the device lasers are unable to penetrate the ocean surface at incidence angles that occur in normal maritime scenarios. Figure 2.23 shows a maritime image and lidar data taken in a similar scenario.

HDL-64E has also been used in surveying. (16) present one such system for road and highway data collection analysis. The system can be used for fast data collection about bridges, road clearances etc and thus eliminating the need to manually measure these structures which not only takes more time and effort but also disturbs normal flow of traffic.

Unrelated to robotics but interestingly a music video has even been made using 3D data from HDL-64E and a structured light system from Geometric Informatics, without the use of any conventional video camera (12).

2.5 Significance of lidar in robotics

Lidar sensors have long been used as exteroceptive sensors in robotics although they are not the only exteroceptive sensors available: Other sensors like sonars, stereovision and radar have also been used extensively. All these exteroceptive sensors are not only used to map the environment around the robot but also for other short-term tasks like obstacle detection, human presence detection, robot localization, etc.

Sonar sensors use sound for ranging purposes so they are slower compared to vision and lidar sensors which lead to a decrease in their usage except in underwater robotics where they are still used. Stereovision sensors are very attractive in robotics because they are light-weigh, low-cost, have low power consumption and are easily available. These properties have lead to their extensive usage in robotics, but they also have some disadvantages however. The accuracy of stereovision sensors decreases with the square of increase in scene depth. This property in turn limits the efficacy of stereovision sensors in mapping applications especially in outdoor environments where the distances to be measured are typically longer compared to the indoor office or home environments. Also, lighting conditions and lack of texture can hinder stereovision.

Unlike stereovision sensors, lidars have the ability of acquiring accurate range measurements at short as well as longer distances. This property, and the fact that lidars are not affected by light and texture conditions, are probably the main reasons why they have continuously been used in robotics. The introduction of a multi-beam lidar prototype by the Velodyne $TeamDAD$ in DARPA Grand Challenge 2005 and its commercialization in 2006 has redefined the way outdoor autonomous robots perceived their environments, as confirmed by its successful exploitation in the 2007 DARPA Urban Challenge. The potential of multi-beam lidar sensors to be employed in outdoor mobile robotics is apparent from the numerous Velodyne HDL-64E applications presented in section 2.4.

One of the challenges in the usage of multi-beam lidar devices in robotics is their high cost. The introduction of multi-beam lidar devices simpler than the HDL-64E addresses this issue to a certain extent. One such device is the Velodyne HDL-32E which does not only cost less but is also lighter and has less laser-beams. (76) introduce the design and calibration of a multi-beam lidar system that employs three SICK LMS-151 lasers fitted on a rotating unit.

Radars have also been used as exteroceptive sensors in outdoor robotics because they are more robust to bad weather conditions such as fog, whereas fog can affect the performance of conventional lidar sensors. Note however that multiple echo detection technology such as the one employed is SICK LMS5xx (1) lidar sensors tackle this issue and make the lidar sensors robust to challenging weather conditions.

The advantages that lidar and especially multi-beam lidar devices offer and their successful exploitation in the field of outdoor autonomous robotics over the last few years proves their potential for the field in the years to come.

Chapter 3

Velodyne lidar calibration

Calibration is a crucial aspect of any sensor system, as its performance highly depends on the accuracy of the calibration. A well calibrated sensor can indeed be exploited to its full capacity whereas a bad calibration can greatly compromise the utility of the sensor. Calibration becomes even more crucial when it comes to the autonomy of any type in a given system: an autonomous system independent of human control relies on its sensors making the sensor performance a critical factor in system performance, success and efficacy of any given task. Exteroceptive sensors in robotics are at the heart of robot knowledge about its environment, which is the essential information for the robot to move around or act in the environment. For example, the data from well calibrated 3D exteroceptive sensors like lidars can be used to extract point, line or planar features from the environment, whereas extraction of these features can be difficult and unreliable or even impossible if the sensor is badly calibrated. Feature detection and repeatability is very important in many SLAM (simultaneous localization and mapping) systems and wrong matching of detected features can lead to a total failure of the SLAM system. Similarly, imprecise calibration can result in inaccurate digital terrain maps and thus erroneous interpretations of the sensed terrain, unreliable obstacle detection, etc.

This chapter presents the technique we developed for the calibration of multi-beam lidar systems. The calibration method is presented along with its implementation for the intrinsic calibration of the Velodyne HDL-64E S2.

3.1 Methodology

The principle lying behind the proposed calibration technique is the comparison of data acquired from the lidar system and a ground truth to estimate the optimal values of the calibration parameters thanks to an optimization technique. This process includes defining a geometric model for the sensor, defining a calibration environment to enable acquisition of scan data, defining a cost function to compare scan data and the ground truth and finally performing optimization using the defined cost function. These steps involved in the calibration process are depicted below.

3.1.1 Sensor modeling

A very basic requirement for the calibration process is the system modeling. This includes defining the internal or external geometry and other physical characteristics of the system. In the intrinsic calibration of cameras for instance, this step corresponds to the choice of a pin-hole or any other model to represent the internal geometry of a camera as explained section 2.2.

A multi-beam lidar system essentially consists of several laser beams. A laser beam can be thought of as a half-ray originating from the laser emitter and extending to infinity. In order to calibrate a lidar system, a set of parameters representing the position and orientation of each laser beam has to be chosen. This choice depends on the physical setting of the system at hand, but also on practical issues, like a priori availability of coarse calibration data for the system: if one has a coarse calibration for the device, one might decide to stick to the parametrization used by this coarse calibration and improve it. Lets define this set of parameters by $\{M_1, \ldots, M_n\}$.

At least five parameters are required to define one laser beam in a 3D coordinate frame: two angles to define the direction of the associated line and three parameters to define the origin point of the beam. If an additive and proportional distance correction factor is required to correct the measurement made by the laser beam, the number of calibration parameters goes to seven per laser beam.

In practice however the exact location of the laser beam origin is not required and therefore the additive distance correction factor can be incorporated into the three parameters defining the position of laser beam. For example, to incorporate an additive distance correction factor of 1m, the point of origin of laser beam can be considered shifted 1m along the laser beam.

Another possibility to parametrize a laser beam is by using two angles and two distance parameters defining an infinitely long line. A fifth parameter, an additive distance correction factor completes the knowledge for converting a raw lidar measurement to a 3D point.

3.1.2 Calibration environment

The purpose of the calibration environment is to define a ground truth with respect to which the acquired data is compared. The selection of a suitable calibration environment depends on a number of factors:

- The environment should be "rich" enough to enable the calibration of the system at hand. For example a simple planar surface might be suitable for calibration of the orientation of a laser beam but will not suffice for the estimation of its position. The scenario changes if the edges of the scanned planar surface are also taken into account: by introducing the fact that the edges must correspond to straight lines, we can also optimize the position parameters of a laser beam.

- The selection of the calibration environment also depends on practicalities like simplicity in its structure and construction. Simplicity in structure implies that it is easy measure or extract ground truth data for the environment, whereas simplicity in construction is a merely practical issue and implies that the environment should be simple and easy to construct in a short time with minimum human input.

- The environment should allow the definition of an objective/cost function which can serve as a quantitative measure between the scan data and the ground truth.

3.1.3 Data segmentation

The calibration environment must contain geometrical objects that are used in calibration, but it also might contain structures that are not useful. Data segmentation consists of extracting, from the acquired scan data, the data that will actually be used during calibration. This data corresponds to the objects for which the ground truth is known, enabling this data to be employed in the calibration process.

For example figure 2.15 (bottom) shows the top view of a region in a room. For the three corner regions in the figure, it is very hard to segment the data, *i.e.* to associate the data point to one or the other wall. The environment chosen for the calibration process should be designed and made to ease the segmentation of data.

3.1.4 Optimization objective function

As mentioned above, an objective function C has to be defined that can provide a quantitative measure of similarity between the scan data and the ground truth. This function forms the basis of the optimization process.

C should provide higher costs if there is more difference between the acquired 3D data and ground-truth environment, and lower costs as the match between acquired 3D scan data and real environment improves.

Another important requirement for C is that it should be sensitive to each of the parameters M to be optimized. It means that the cost provided by this function must vary with the values of the parameters to be optimized, and of course this increase or decrease in the cost should be in accordance with the first criteria for C *i.e.* the cost should be lower for better matches between the acquired 3D data and ground truth and vice versa. The suitability of the cost function to perform a successful calibration process can be analyzed by computing the partial derivatives of the cost function with respect to the parameters to be optimized. In order for the cost function to be suitable, these partial derivatives should generally be non-zero and non-constant.

$$\partial C / \partial M_i \neq constant \tag{3.1}$$

3.1.5 Optimization

Optimization is the estimation of the optimal value(s) of one or more variables, where “optimal” means the values that minimize or maximize a given function (known as the “objective function”) as desired. If the values to be estimated are known to lie is a given range (typically governed by a set of inequalities) the problem is known as constrained optimization.

Typically the optimization problems are solved by iterative methods that evaluate either only function values or function values along with their Hessians or gradients iteratively in order to estimate the optimal values for the variables to be estimated. Examples of these methods include Newton's method, Sequential quadratic programming, etc.

The optimization process is the main calibration step. Using the calibration objective function the calibration parameters M are optimized by minimizing the difference between scans and ground truth. The optimization process during this step can be formulated as a constrained or unconstrained problem depending on whether the calibration parameters are expected to lie in a given range or not. And then the optimization problem is solved using a suitable mathematical technique among the ones available.

3.2 Implementation

The proposed calibration technique was implemented on a real multi-beam lidar device. The device is the Velodyne HDL-64E S2 presented in detain in section 2.3.

This sections details different steps involved in the implementation including the device model, calibration environment, the chosen objective function, and it's suitability for the calibration and the optimization process.

3.2.1 Geometric model

The geometric model defined in (90) was chosen for calibration implementation for the Velodyne HDL-64E S2. The main reason for choosing this model for the implementation of the calibration technique is the availability of coarse calibration data that was provided by the manufacturer along with the device. This geometric model is depicted below.

64 lasers of the lidar device can be thought of as a set of 64 half-rays. Each half-ray defines the position and orientation of the respective laser beam with respect to a sensor-fixed coordinate frame. Each half-ray is specified by five parameters which in turn are required to convert a raw laser return to a 3D point represented in the sensor-fixed frame. These five parameters are briefly described below and their usage is explained further.

Distance Correction D_{corr}: Each laser has a specific value that needs to be added to the raw distance D_{ret} value returned by the laser in order to get the correct distance D of the observed point from the laser origin. This correction value is D_{corr}. As observed by (71) and (78) a proportional distance correction is necessary for the previous Velodyne model HDL-64E, but not for the current S2 version of the device, both according to the device documentation and to our experimentation.

Vertical Offset V_o: This is the distance measured orthogonally to the laser beam, representing the distance of laser beam from the origin in a vertical sense. The segment OA represents this value in figure 3.1 where O is the origin of sensor-fixed frame.

Horizontal Offset H_o: This value is the horizontal counterpart of Vertical Offset. The segment OB represents this in figure 3.2.

Vertical Correction Angle θ: This value represents how the 64 lasers are distributed vertically to cover the 27 degrees vertical field of view. It is the angle made by the laser beam with the $x-y$ plane, as shown in figure 3.1. *E.g.* for a laser beam lying in the $y-z$ plane, this would be the angle made by laser beam around the x axis. Seen from the back of each laser point of origin, positive values of θ represent the deviation of lasers upwards and vice versa.

Rotational Correction Angle α: This is the rotational counterpart of the Vertical Correction Angle, *i.e.* the angle made by the laser beam with the $y-z$ plane. Seen from the back of each laser point of origin, positive values of α represent the deviation of lasers toward left and vice versa.

As the lidar head spins, its current rotational angle is denoted as ϕ. Each laser has a different value of α so an angle β can be defined such that $\beta = \phi - \alpha$. In this way β represents the orientation of laser beam relative to the current rotational angle of spinning lidar head.

Figure 3.1 shows the side view of the lidar. The blue rectangle represents the lidar head, and O represents the origin of the fixed coordinate system of the lidar (physically this origin is located at the bottom of lidar base but for the sake of simplicity it is shown on the lidar head in the figure).

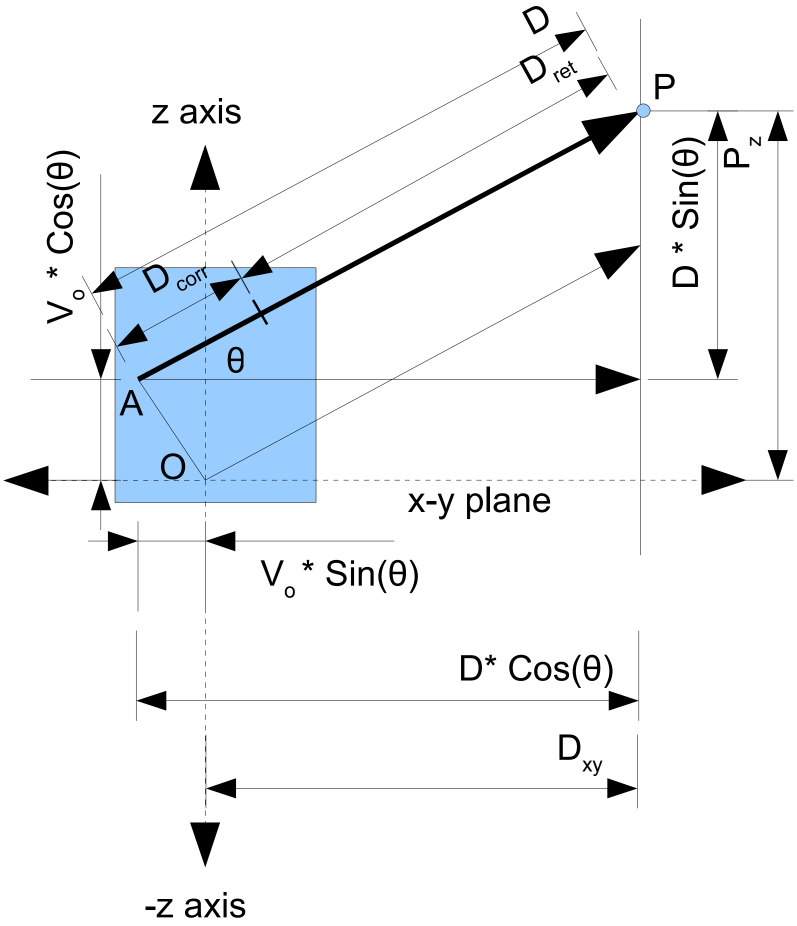

Figure 3.1: Side view of the lidar.

As mentioned earlier, segment OA represents the vertical offset value for the laser. The laser is emitted and it hits the point P, and a distance value is returned by the lidar for this point. D_{corr} is added to this distance value to get the actual distance between points A and P. This distance is denoted by D.

$$D = D_{ret} + D_{corr} \tag{3.2}$$

The z-coordinate of point P can be calculated as follows:

$$P_z = D * \sin(\theta) + V_o * \cos(\theta) \tag{3.3}$$

Figure 3.1 also shows D_{xy} which can be represented as:

$$D_{xy} = D * \cos(\theta) - V_o * \sin(\theta) \tag{3.4}$$

D_{xy} will now further be used to find the x and y coordinates of point P.

Figure 3.2 shows the lidar head as seen from top. Segment OB represents the horizontal offset for the laser. P_{xy} is the projection of point P in the $x - y$ plane and is at a distance of D_{xy} from point B. In this case the rotational correction angle for the laser is positive as the laser beam (whose projection in $x - y$ plane is seen here as line BP_{xy}) is oriented leftwards as seen from the back of the laser point of origin. For the sake of simplicity, the figure shows the case with the lidar head rotational angle equal to zero, which in turn makes $\beta = -\alpha$. From the figure it can be seen that:

$$P_x = D_{xy} * \sin(\beta) - H_o * \cos(\beta) \tag{3.5}$$

Note that in this case β is negative and P_x is also a negative value as point P lies towards the $-x$ axis. Similarly it can be seen that:

$$P_y = D_{xy} * \cos(\beta) + H_o * \sin(\beta) \tag{3.6}$$

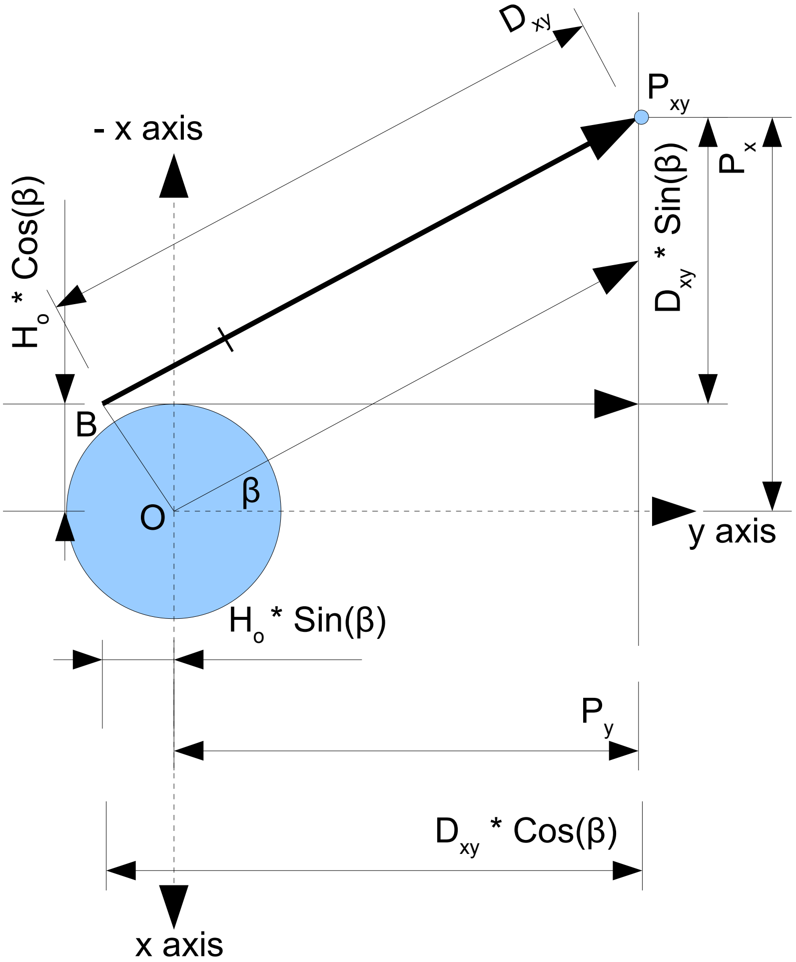

Figure 3.2: A top view of lidar.

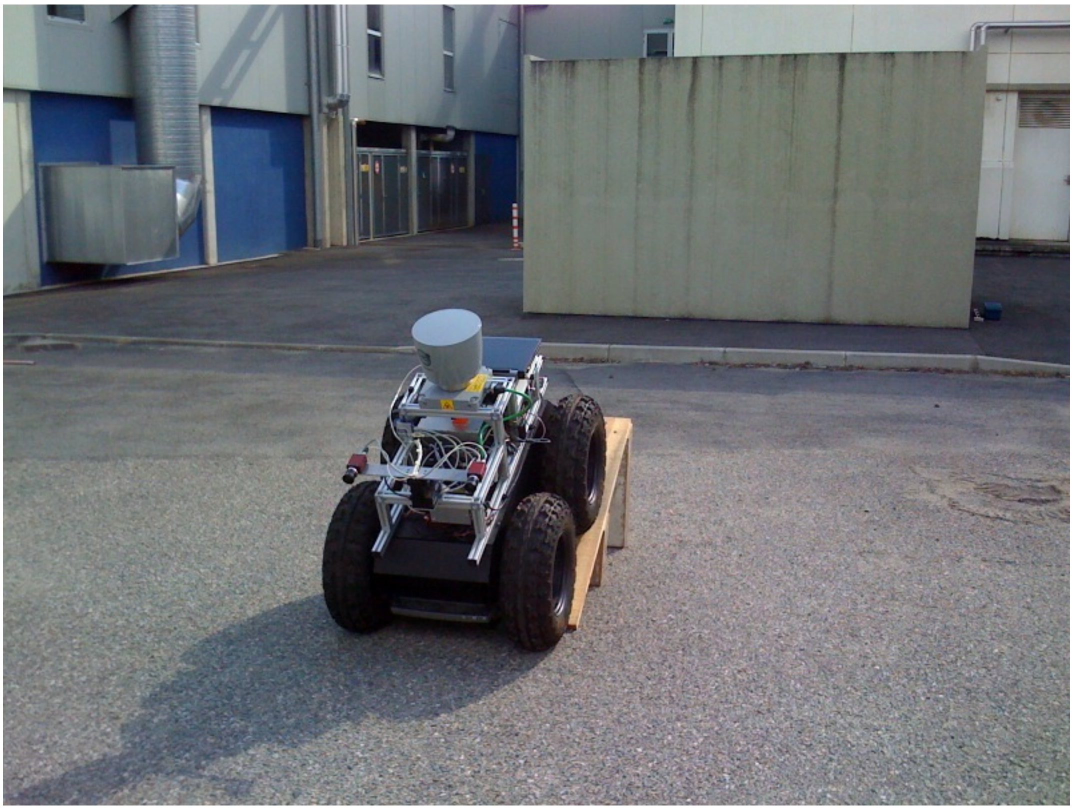

Figure 3.3: Robot Mana on the wooden ramp scanning the calibration target

3.2.2 Calibration environment

Keeping in view the environment selection issues discussed in subsection 3.1.2, a 4.40m wide planar wall was used as the calibration environment. The wall was scanned by placing the lidar at nine locations ranging from 3 to 22m from the lidar device. The distribution of scan data over a range of distances is very important. Sufficiently distributed data is necessary to ensure the estimation of calibration parameters to be independent of any bias on a specific distance.

At each location five scans were taken with the sensor titled at different elevation angles in order to get the planar scan data for all the 64 laser beams of the device. The robot was tilted by moving it on a wooded ramp as shown in the figure 3.3. The figure also shows the planar wall that was used as the calibration target. Figure 3.4 shows three scans of the calibration target superimposed in a single plot.

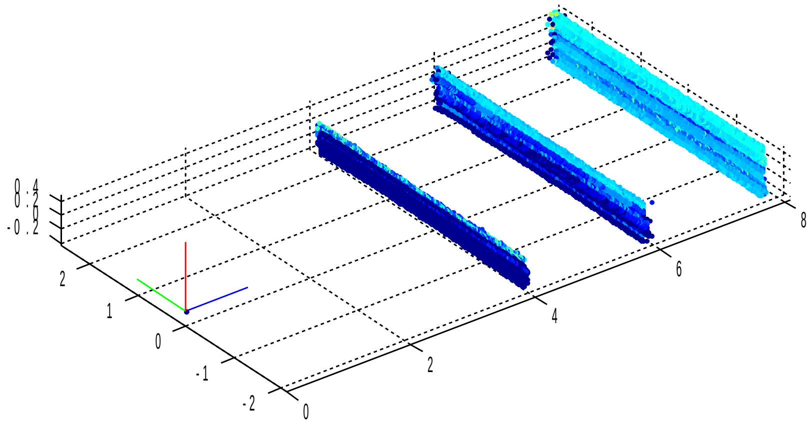

Figure 3.4: Three scans of the calibration target

3.2.3 Objective/Cost function

If it is possible to accurately align the planar calibration target to the axes of $sensor-fixedframe$, the cost function C can be defined as the variance of 3D data along the plane normal: if the calibration target is scanned with the $y-axis$ of the $sensor-fixedframe$ aligned to the calibration target, the cost can be the average squared difference between the $x-coordinates$ of each point and the x_{mean}. A similar cost value can be defined if the calibration target is aligned to the $x-axis$ of the $sensor-fixedframe$ by taking into account the $y-coordinates$ of the scanned points.

In practice it is hard to accurately align the plane with the x or y of the $sensor-fixedframe$. The cost function C was therefore chosen to be the sum of squared perpendicular-distances of all points in a planar scan (of the calibration target) to a plane that best fitted the scan, divided by the total number of points forming the plane. The plane fitting was performed using principal component analysis (PCA) in Matlab. C can be mathematically represented as:

$$C = \Sigma(D_{i,Perp})^2/n \tag{3.7}$$

where n is the total number of points in current scan of the plane (the calibration target).

3.2.4 Suitability Analysis

As mentioned in section 3.1.4, the suitability of the cost function for the optimization depends on the cost function sensitivity to the variation of the parameters to be estimated. As our chosen cost function depends on the distances of x, y and z coordinates of 3D data, the suitability of chosen cost function can be ensured by finding the partial derivatives of P_x, P_y and P_z with respect to each of the three calibration parameters to be optimized, *i.e.* D_{corr}, θ and α.

Using equations 3.5, 3.6 and 3.3, the partial derivatives with respect to D_{corr} can be computed as follows:

$$\begin{aligned}
\partial P_x/\partial D_{corr} &= \partial(D_{xy} * \sin\beta - H_o * \cos\beta)/\partial D_{corr} \\
&= \partial(((D_{ret} + D_{corr}) * \cos\theta - V_o * \sin\theta) * \sin\beta - H_o * \cos\beta)/\partial D_{corr} \\
&= \partial((D_{ret} + D_{corr}) * \cos\theta \sin\beta)/\partial D_{corr} \\
&= \partial(D_{corr} * \cos\theta \sin\beta)/\partial D_{corr} \\
&= \cos\theta \sin\beta
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
\partial P_y/\partial D_{corr} &= \partial(D_{xy} * \cos\beta + H_o * \sin\beta)/\partial D_{corr} \\
&= \partial(((D_{ret} + D_{corr}) * \cos\theta - V_o * \sin\theta) * \cos\beta + H_o * \sin\beta)/\partial D_{corr} \\
&= \partial((D_{ret} + D_{corr}) * \cos\theta \cos\beta)/\partial D_{corr} \\
&= \partial(D_{corr} * \cos\theta \cos\beta)/\partial D_{corr} \\
&= \cos\theta \cos\beta
\end{aligned} \tag{3.9}$$

$$\begin{aligned}
\partial P_z/\partial D_{corr} &= \partial(D * \sin\theta + V_o * \cos\theta)/\partial D_{corr} \\
&= \partial((D_{ret} + D_{corr}) * \sin\theta + V_o * \cos\theta)/\partial D_{corr} \\
&= \partial((D_{ret} + D_{corr}) * \sin\theta)/\partial D_{corr} \\
&= \sin\theta
\end{aligned} \tag{3.10}$$

From $\partial P_x/\partial D_{corr}$ it is clear that the conditions that make the partial derivative equal to zero are $\theta = 90°$ and $\beta = 0°$. This makes intuitive sense as $\theta = 90°$ means that the laser is pointing upwards and in such a situation it is impossible to scan a plane (calibration target) which is parallel to the laser. Similarly, as β defines the current orientation of a laser beam, $\beta = 0°$ means that the laser is parallel to $y - axis$ of lidar frame and therefore any variation in D_{corr} would not affect P_x for the point being viewed as β remains 0°. In the case of Velodyne HDL-64E S2, as the lidar head is constantly rotating the value of β is constantly changing, and moreover the values for θ for all lasers in the lidar are much smaller than 90°.

Similarly, the conditions that make $\partial P_y/\partial D_{corr}$ equal to zero are $\theta = 90°$ and $\beta = 90°$. As with the previous case, the condition $\beta = 90°$ is not a problem because the lidar is constantly rotating as we acquire the data. The condition that makes $\partial P_z/\partial D_{corr}$ equal to zero is $\theta = 0°$. This would mean that for lasers with zero pitch angle, the z coordinates of data points will not play any role in the optimization process. This does not pose any problem because the optimization process is based on 3D data and not only on the z coordinates of data. Moreover for the system at hand, the pitch angle for any laser beam is not exactly zero. This analysis leads to the conclusion that our chosen cost function is suitable to be used for the estimation of D_{corr} using optimization.

Similarly using equations 3.5, 3.6 and 3.3, the partial derivatives of P_x, P_y and P_z with respect to θ can be computed as follows:

$$\begin{aligned}
\partial P_x/\partial\theta &= \partial(D_{xy} * \sin\beta - H_o * \cos\beta)/\partial\theta \\
&= \partial((D * \cos\theta - V_o * \sin\theta) * \sin\beta - H_o * \cos\beta)/\partial\theta \\
&= \partial(D * \cos\theta\sin\beta - V_o * \sin\theta\sin\beta)/\partial\theta \\
&= D * (-\sin\theta)\sin\beta - V_o * \cos\theta\sin\beta \\
&= -D\sin\beta\sin\theta - V_o\sin\beta\cos\theta \qquad (3.11)
\end{aligned}$$

$$\begin{aligned}\partial P_y/\partial\theta &= \partial(D_{xy} * \cos\beta + H_o * \sin\beta)/\partial\theta \\ &= \partial((D * \cos\theta - V_o * \sin\theta) * \cos\beta + H_o * \sin\beta)/\partial\theta \\ &= \partial(D * \cos\theta\cos\beta - V_o * \sin\theta\cos\beta)/\partial\theta \\ &= D * (-\sin\theta)\cos\beta - V_o * \cos\theta\cos\beta \\ &= -D\sin\theta\cos\beta - V_o\cos\theta\cos\beta \end{aligned} \tag{3.12}$$

$$\begin{aligned}\partial P_z/\partial\theta &= \partial(D * \sin\theta + V_o * \cos\theta)/\partial\theta \\ &= -D\cos\theta - V_o\sin\theta \end{aligned} \tag{3.13}$$

The conditions which make $\partial P_x/\partial\theta$ and $\partial P_y/\partial\theta$ equal to zero are $\beta = 0°$ and $\beta = 90°$ respectively. As with the case for D_{corr}, for the constantly rotating lidar, these partial derivatives remain non-zero for the type of 3D datasets being used and therefore the chosen cost function is suitable to be used for the estimation of θ using optimization.

Similarly the partial derivatives of P_x, P_y and P_z with respect to α can be computed as follows:

$$\begin{aligned}\partial P_x/\partial\alpha &= \partial(D_{xy} * \sin\beta - H_o * \cos\beta)/\partial\alpha \\ &= \partial(D_{xy} * \sin(\phi-\alpha) - H_o * \cos(\phi-\alpha))/\partial\alpha \\ &= \partial(D_{xy} * (\sin\phi\cos\alpha - \cos\phi\sin\alpha) - H_o * (\cos\phi\cos\alpha + \sin\phi\sin\alpha))/\partial\alpha \\ &= (D * \cos\theta - V_o * \sin\theta) * (-\sin\phi\sin\alpha - \cos\phi\cos\alpha) \\ &\quad - H_o * (-\cos\phi\sin\alpha + \sin\phi\cos\alpha) \end{aligned} \tag{3.14}$$

$$\begin{aligned}\partial P_y/\partial\alpha &= \partial(D_{xy} * \cos\beta + H_o * \sin\beta)/\partial\alpha \\ &= \partial(D_{xy} * \cos(\phi-\alpha) + H_o * \sin(\phi-\alpha))/\partial\alpha \\ &= \partial(D_{xy} * (\cos\phi\cos\alpha + \sin\phi\sin\alpha) + H_o * (\sin\phi\cos\alpha - \cos\phi\sin\alpha))/\partial\alpha \\ &= (D * \cos\theta - V_o * \sin\theta) * (-\cos\phi\sin\alpha + \sin\phi\cos\alpha) \\ &\quad + H_o * (-\sin\phi\sin\alpha - \cos\phi\cos\alpha) \end{aligned} \tag{3.15}$$

$$\begin{aligned}\partial P_z/\partial\alpha &= \partial(D * \sin\theta + V_o * \cos\theta)/\partial\alpha \\ &= 0 \end{aligned} \tag{3.16}$$

There are no conditions that make $\partial P_x/\partial\alpha$ and $\partial P_y/\partial\alpha$ equal to zero. $\partial P_z/\partial\alpha$ however is zero. This makes intuitive sense, because a change in the rotational correction angle α will not change the $z-coordinate$ of the acquired 3D point. In other words if a laser beam points slightly leftwards or rightwards, only the x and y coordinates of the acquired 3D point will change. It suggests that the chosen cost function is suitable for the estimation of α but only x and y coordinates of 3D points forming the calibration target will contribute to the estimation.

3.2.5 Optimization

The optimization process was implemented using Matlab function $fmincon$ (49). The function uses sequential quadratic programming to solve a quadratic programming problem at each iteration.

The computational cost of optimization process is not of a great concern because the process is done offline and has to be done only once to calibrate the device. The optimization process took several hours to complete on a normal laptop machine.

3.3 Results

This section presents some results from the calibration implementation presented in section 3.2. Among the five calibration parameters for the Velodyne HDL-64E S2, D_{corr}, α and θ are the most important. This is because of the fact that errors induced by a bad estimate of these parameters in the precision of 3D coordinates of acquired data is magnified with the increase in distance of the scanned object. The errors induced by imprecise calibration of V_o and H_o on the other hand do not depend on the variation in the distance of scanned object. Therefore in the current implementation only D_{corr}, α and θ were calibrated for the HDL-64E S2 device at hand. The default calibration data provided by the manufacturer along with the device was used as the initial estimates of the parameters to be estimated during the optimization step of the calibration process.

Table 3.1: Standard deviations (in meters) in depth of planar data

	Default Calibration	**Recalibration**
4m	0.0234	0.0215
6m	0.0326	0.0291
8m	0.0170	0.0116
10m	0.0168	0.0105
12m	0.0186	0.0119
14m	0.0187	0.0128

3.3.1 Recalibrating a subset of lasers

In a first step only 16 of the 64 lasers having the low (near zero) pitch angles were calibrated in order to validate the proposed calibration technique on a smaller scale before calibrating all the 64 lasers in the device. This was done using the scans of the calibration target taken at 2m steps between distance of 4m to 14m between the lidar device and the calibration target. At each location the calibration target was scanned by first roughly aligning it to the $x-axis$ and then to the $y-axis$ of the lidar sensor-fixed frame. Figure 3.5 presents a result of improved calibration resulting from this step. At top, the figure shows the scan of the rear of a vehicle computed using default calibration data and at bottom it shows the same scan using optimized calibration parameters. While the figure shows a qualitative comparison of the improvement in calibration it is also important to quantitatively compare the default and new calibration data. One way of doing it is by measuring the standard deviation of data forming a scanned plane along the depth of the plane. Table 3.1 shows the improvement in standard deviations in depth for a subset of planar data used in the recalibration process. Table 3.2 shows the improvements in standard deviation in depth of some planar data that was not used during the optimization step. The improvements in the standard deviation of planar data and qualitative improvement shown in 3.5 validate the proposed calibration technique.

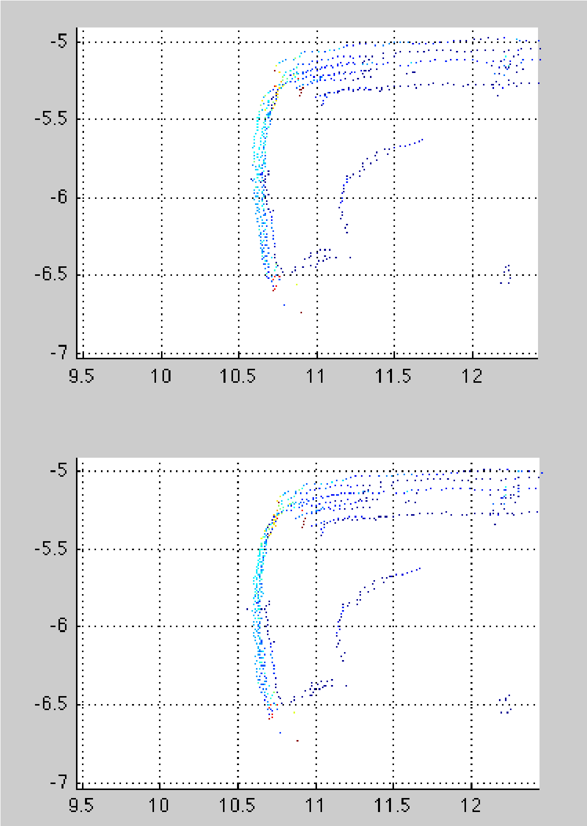

Figure 3.5: Comparing default (top) and optimized (bottom) calibration parameters for 16 laser beams with low pitch angles. Scans of rear of a vehicle

3.3.2 Recalibrating all 64 lasers

After the initial validation of the proposed calibration technique on 16 lasers, the technique was applied to recalibrate the D_{corr}, α and θ parameters for all the 64 lasers in the Velodyne HDL-64E S2 device simultaneously using the acquired scans of the calibration target as explained in subsection 3.2.2.

Table 3.3 shows a comparison of the sum of squared differences error for different scans of the calibration target computed with respect to the an ideal plane filled to the corresponding scan of the calibration target. A marked decrease in the error can be observed not only in the data used during optimization i.e. the scans taken from 3m to 22m distances but also for the scans taken at 26m distance that were not used in the optimization process.

To qualitatively analyze and compare the default and recalibrated parameters, dif-

Table 3.2: Standard deviations (in meters) for data not used in optimization

	Default Calibration	**Recalibration**
16m	0.0210	0.0132
18m	0.0221	0.0142

Table 3.3: Sum of squared distance error for the scan points forming the plane

	Elevation 1		**Elevation 2**		**Elevation 3**		**Elevation 4**		**Elevation 5**	
	Def	**Recal**	**Def**	**Recal**	**Def**	**Recal**	**Def**	**Recal**	**Def**	**Recal**
3m	1.220	0.980	1.517	0.805	1.396	0.871	1.354	0.807	1.352	0.874
5m	1.867	0.939	1.178	0.687	1.051	0.488	0.944	0.503	0.792	0.485
6.7m	0.969	0.236	0.844	0.310	1.005	0.291	0.921	0.172	0.83 5	0.159
8.5m	0.478	0.230	0.663	0.324	0.856	0.184	0.701	0.098	0.70 3	0.088
10m	0.377	0.132	0.743	0.216	0.903	0.182	0.582	0.127	0.506	0.098
12.5m	0.392	0.112	0.954	0.389	0.858	0.160	0.537	0.106	0.4 96	0.077
15m	0.475	0.218	0.838	0.267	0.797	0.133	0.516	0.114	0.536	0.079
18.5m	0.378	0.101	0.732	0.189	0.849	0.089	0.574	0.094	0.7 30	0.056
22m	0.374	0.114	0.982	0.131	0.540	0.087	0.686	0.143	0.547	0.076
26m	0.462	0.152	0.826	0.238	1.161	0.207	0.867	0.134	0.736	0.087

ferent views of 3D scan are presented below. Figure 3.6 shows the 3D view of a scan using default and recalibrated parameters. Figure 3.7 the top view of the same scan. To allow a more in-depth comparison of the default and recalibrated parameters, the zoomed-in views of the areas highlighted by four red circles in 3.7 are presented and analyzed as follows.

Figure 3.8 shows the zoomed-in view of the structure highlighted by circle-1 in the figure 3.7. The structure is the planar wall that was also used as the calibration target in the scans acquired and used during the recalibration process. Figure 3.8 (top) shows the structure using the default geometrical parameters whereas figure 3.8 (bottom) shows the structure using recalibrated geometrical parameters. A marked decrease in the thickness of the planar structure can be seen between the top and the bottom figures. Quantitatively the sum of squared distance error decreased from 3.620 to 1.718 for the structure shown in the figure. The figure also shows a displacement in the exact location of the structure when using the default and recalibrated parameters. While acquiring the scan, the distance of the planar structure from the lidar device origin was measured by hand to be 8.16m. This distance in the shown scans in the figure turns out to be 8.04m and 8.17m for the default and recalibrated parameters respectively. The decrease in thickness and the closer confirmation of the location of the structure to the ground truth measurement indicate the improvement in the device calibration.

Figure 3.9 shows the zoomed-in view of the area highlighted by circle-2 in the figure 3.7. The top figure shows the are using the default geometrical parameters and the bottom figure shows the area using the recalibrated parameters. The figure shows the decrease in thickness for the planar structures being scanned and thus indicating the improvement in geometrical parameter calibration.

Figure 3.10 shows the zoomed-in view of the structure highlighted by circle-3 in the figure 3.7. The planar structure shown in the figure is the face of a small tin-shed. The comparison between the top and bottom figures show a slight decrease in thickness of the planar data and thus as improvement in the device calibration.

Figure 3.11 shows the zoomed-in view of the structure highlighted by circle-4 in the figure 3.7. Unlike the structures shown in figures 3.8, 3.9 and 3.10 the structure shown in the figure 3.11 shows the increase in thickness of the planar walls. Quantitatively the sum of squared distance error for the lower/horizontal planar-wall shown in the figure increases from 1.782 to 3.303.

3.3.2.1 Recalibration with an alternate geometric model

(22) use a slightly different geometric model for defining laser beam position and orientation inside the Velodyne lidar device. Their model differs from the geometric model presented in the subsection 3.2.1 in that the V_o and H_o are not measured orthogonally to the laser beam but instead V_o is the height of the point of origin of the laser beam from the $x-y$ plane, and H_o is the distance of the point of origin of the laser beam from the $x - z$ plane (*cf.* figure 3.12). We also implemented the recalibration for D_{corr}, α and θ parameters for all the 64 laser beams in the Velodyne device using this geometric model and it gave similar recalibration results as our geometric model. Figure 3.13 shows a qualitative comparison of recalibration results using the geometric model from (22).

3.4 Conclusion

The calibration of novel multi-beam lidar devices such as the Velodyne HDL-64E S2 is a complicated task. Such devices have multiple laser beams and each beam has a set of geometric and distance correction parameters to be calibrated. This makes the total number of parameters to be calibrated on the order of a few hundred. This chapter presents a method for the calibration of orientation angles and distance correction factor for such devices by optimizing the values of intrinsic parameters by comparing the acquired point cloud to the ground truth geometry of a planar calibration target. The results show that recalibration improves the acquired 3D point cloud data in terms of the range measurement confirming to the ground truth distance of the objects from the lidar device (*cf.* figure 3.8). The results also show that in most of the areas in the acquired point cloud, the recalibration improves the 3D point cloud (qualitatively as decrease in the thickness of planar structures seen from the top and quantitatively). But in some areas of the acquired point cloud, the recalibration decreases the quality of acquired data in terms of thickness of planar objects as seen from the top. In a nutshell although the recalibration improves confirmation of range measurements to the ground truth distances of objects, and increases the over all point cloud quality, there is still room for improvement in such devices calibration.

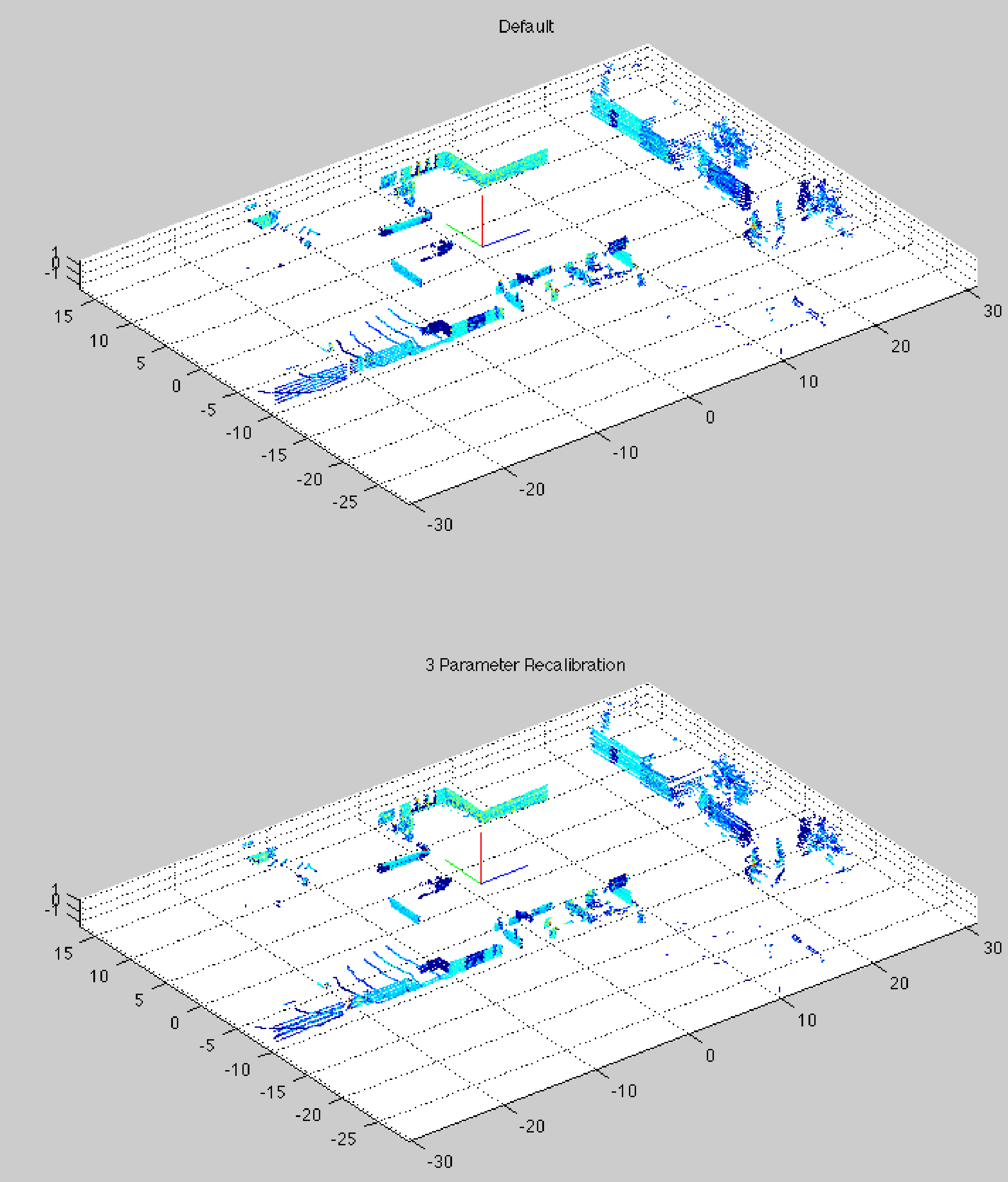

Figure 3.6: Comparing the default and recalibrated parameters. 3D view of a structured outdoor environment. (units in meters)

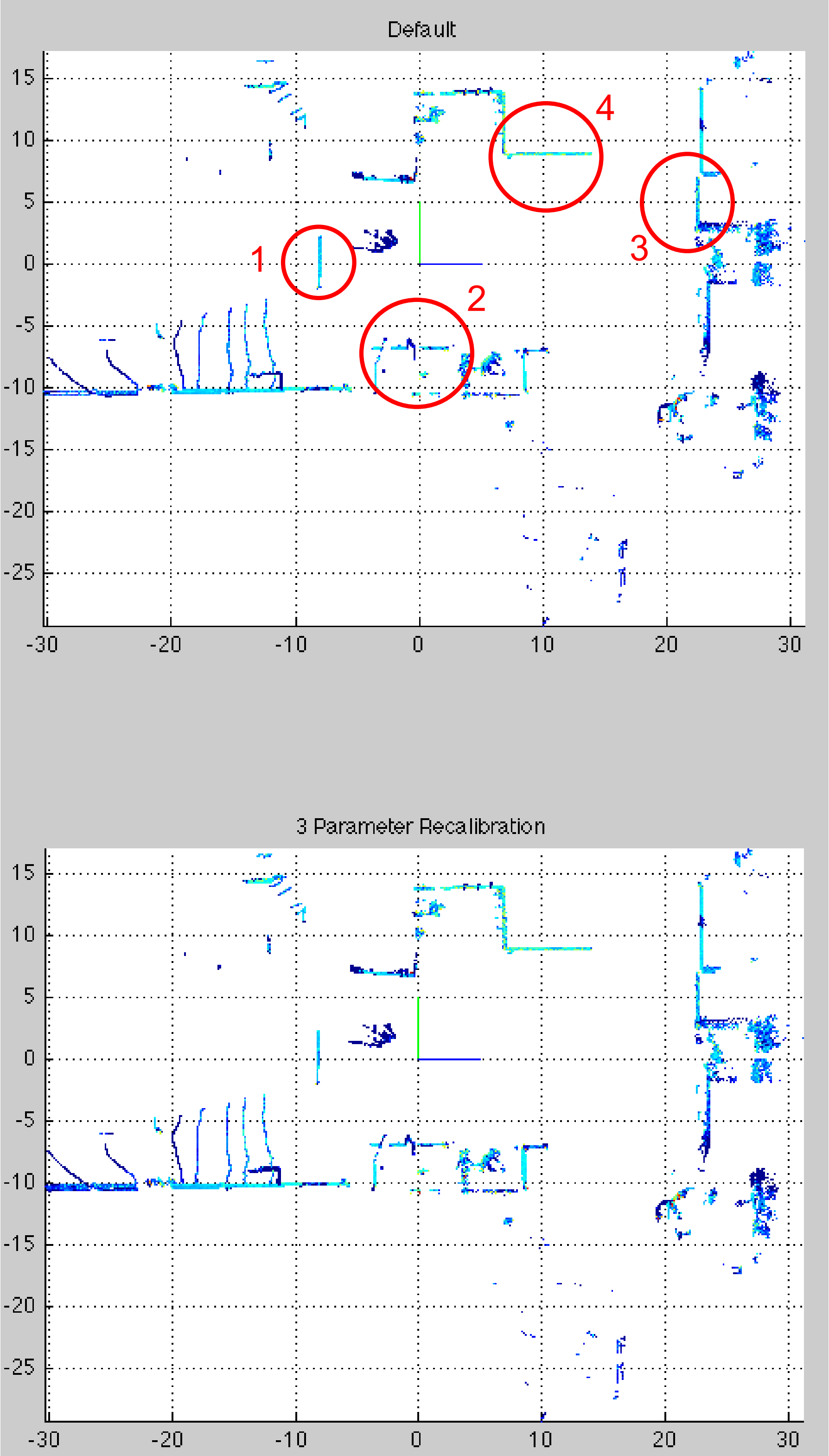

Figure 3.7: Top views of the scan shown in figure 3.6. Units in meters.

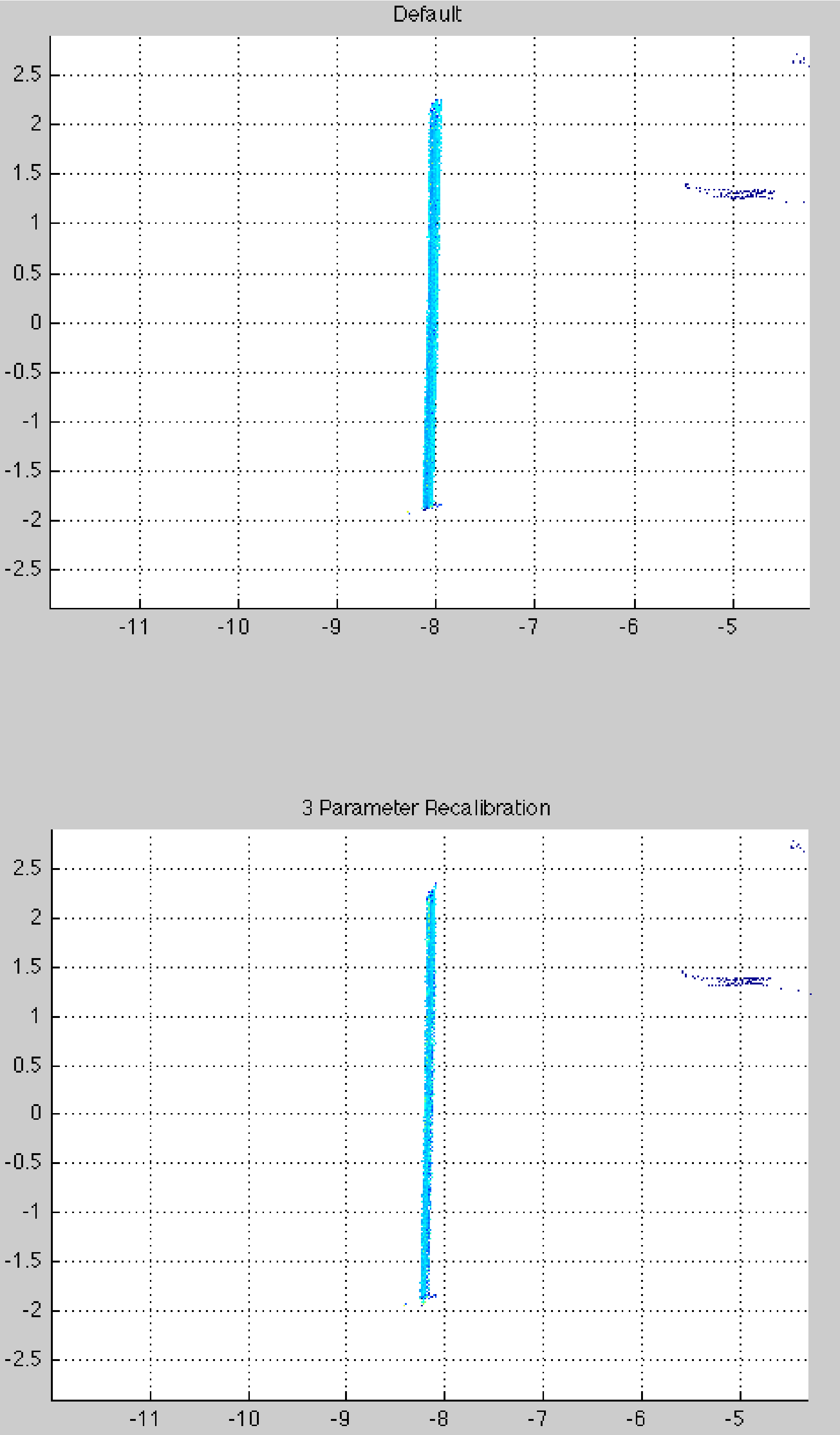

Figure 3.8: Zoomed-in view of the structure highlighted by circle-1 in figure 3.7.

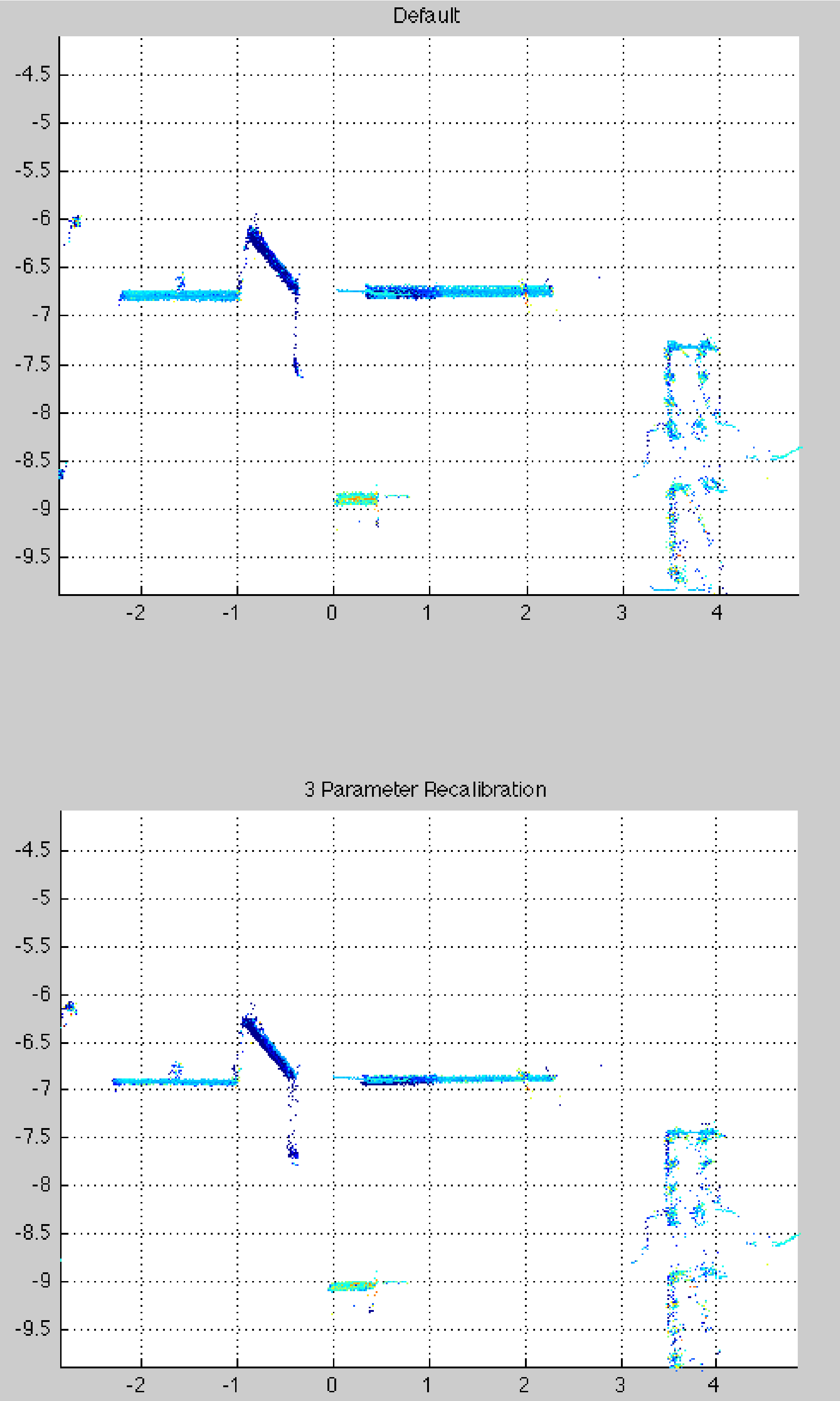

Figure 3.9: Zoomed-in view of the structure highlighted by circle-2 in figure 3.7.

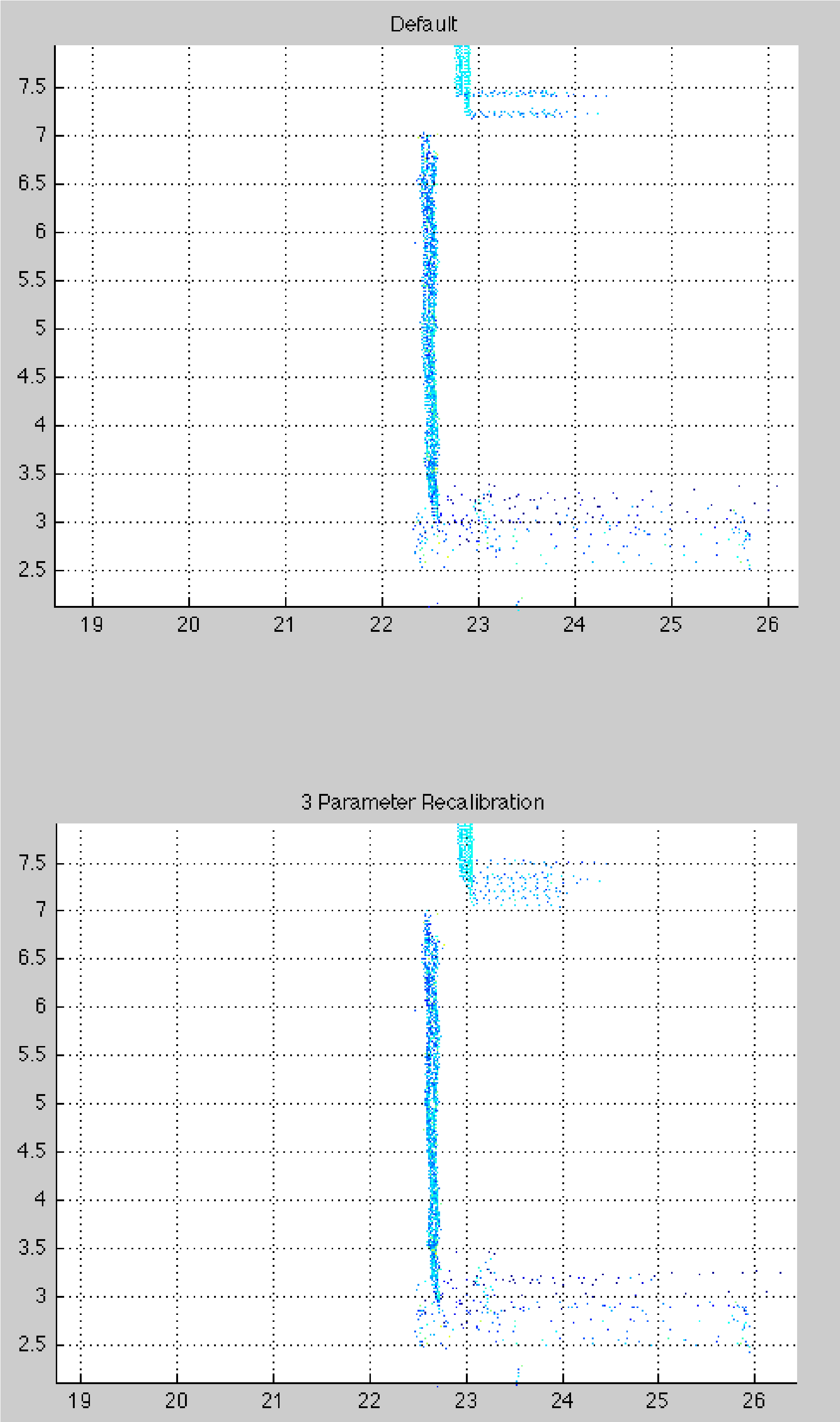

Figure 3.10: Zoomed-in view of the structure highlighted by circle-3 in figure 3.7.

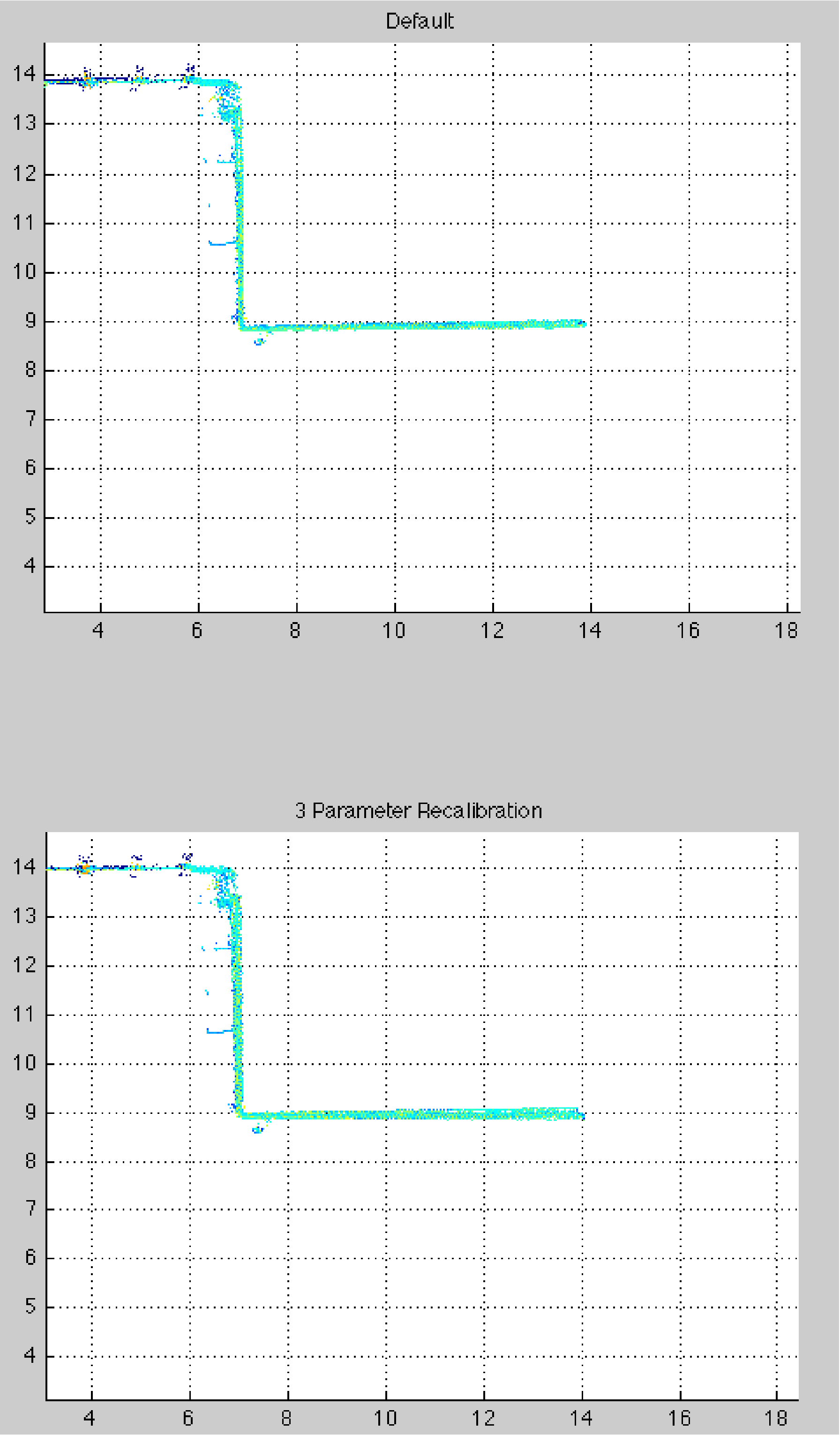

Figure 3.11: Zoomed-in view of the structure highlighted by circle-4 in figure 3.7.

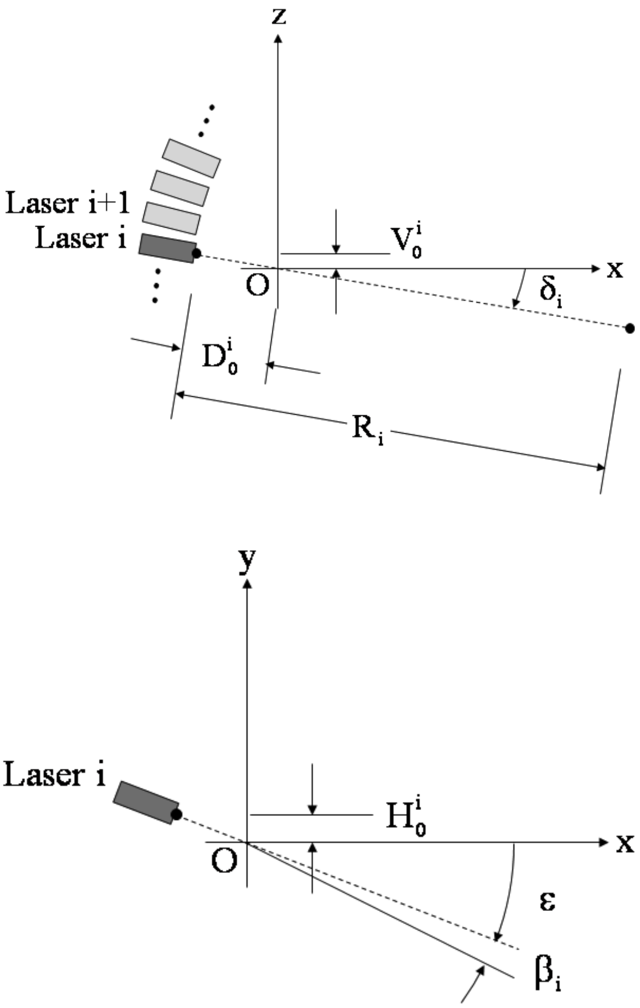

Figure 3.12: Geometric model for Velodyne laser beams from (22): V_o^i and H_o^i represent the Vertical and Horizontal offsets respectively.

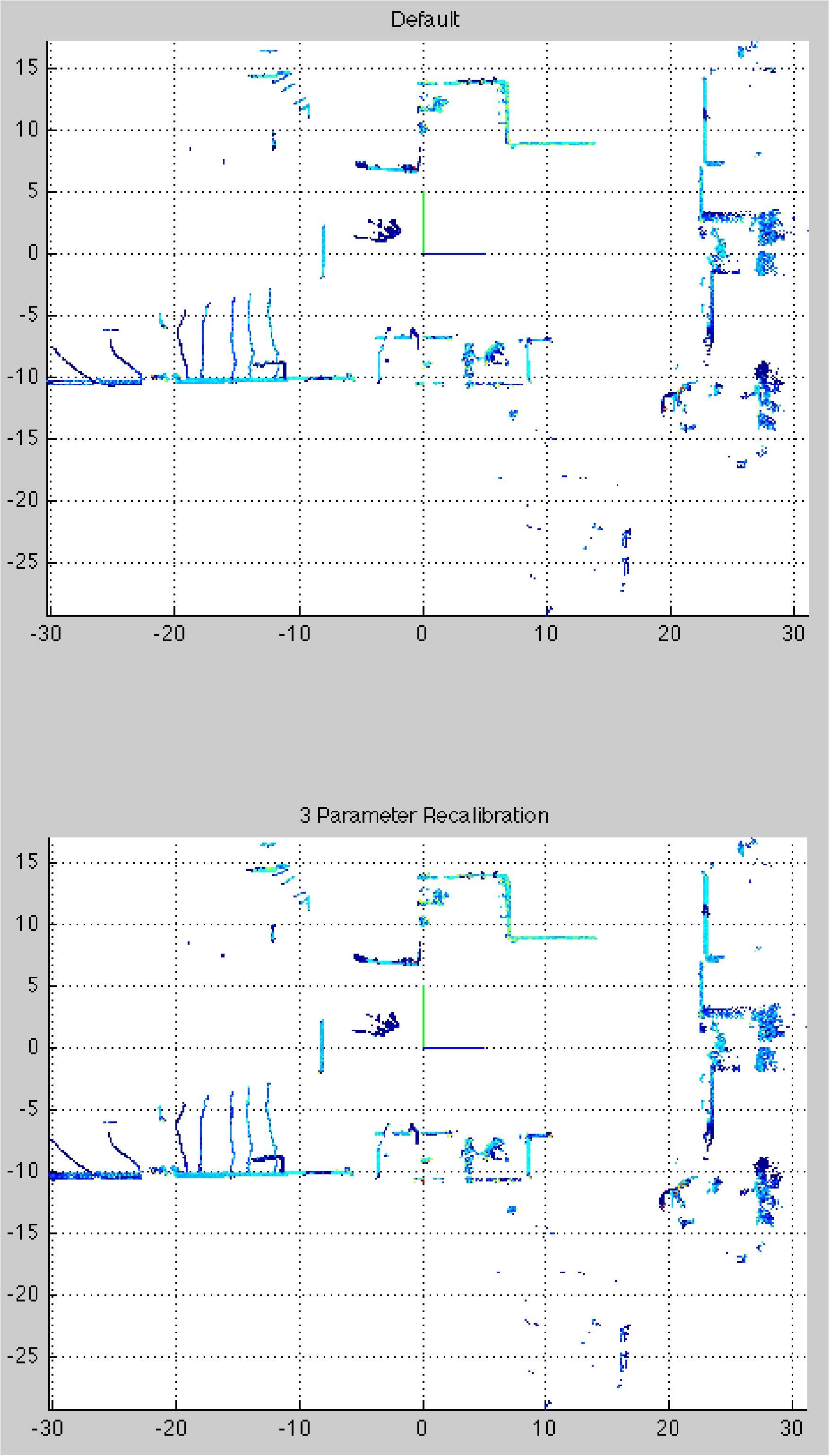

Figure 3.13: Recalibration using the laser-beam geometric model used in (22).

Part II

Qualitative Localization

Chapter 4

The localization problem

"Localization" in a general sense is the determination of location information of an object with respect to a frame of reference. In a mobile robotics context, localization refers to the estimation of robot position and orientation in a reference frame along with an estimate of the uncertainty or error in the position information.

The estimation of uncertainty, or in other words the confidence in location information is very important especially in the case of autonomous robotics. For the autonomous operation of a robot in an environment it is imperative that the robot be able to localize itself within a reasonable error bound, primarily for it's own safety and that of its environment. For instance figure 4.1 shows a scenario where a robot starts to move towards its goal position at time t_1 but at time t_4 the uncertainty in its position estimate is large enough to risk a collision while trying to pass through the open area. In such a situation it is safer for the robot not to proceed after time t_4 unless it can perceive the open passage by using any exteroceptive sensors like camera or lidars.

4.1 Importance of localization, Why localize?

A very basic requirement for an autonomous mobile robot to be able to move around in an a priori known or unknown environment is its ability to correctly localize itself.

Often in mobile robotics, robot missions are expressed in terms of localization. A simple example is the classical "go-to (x,y)" task in which a robot has to move to the position expressed by the coordinates x and y to perform any given operation there. For example a servant robot moving indoors might need to go to a specific position

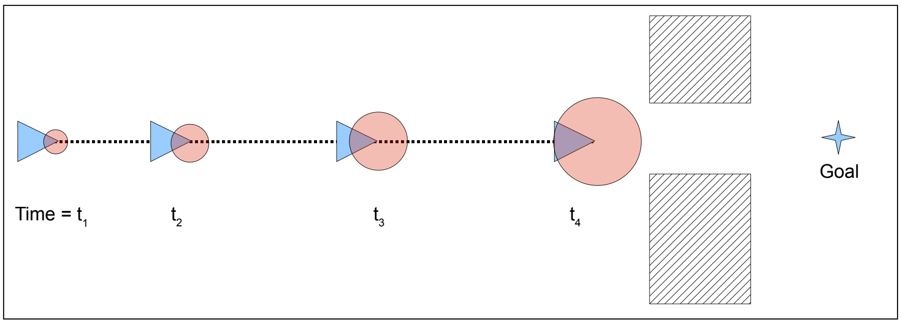

Figure 4.1: Importance of the estimation of error in localization: the robot must not move forward after time t_4 for safety as the uncertainty in robot position (shown as red circles) is larger than the width of the available space to pass through in order to reach the goal position.

where the refrigerator is placed in order to check for the presence of a certain drink, or simply go to an electric socket to plug itself in for charging. Similarly an outdoor mobile robot might need to go to a certain location to deploy or collect some material. In a similar manner a robot mission might involve following a set of way-points or following a predefined trajectory in addition to achieving a final goal location. For example if a team of underwater robots is working collaboratively to explore the seabed, each robot might need to follow a specific trajectory in order to communicate with other underwater robots at predefined rendezvous locations.

Often during autonomous operation, a robot does not know its environment a priori. In such cases the robot might have to build a model of its environment for performing basic tasks such as obstacle avoidance and planning a path to a goal location. Building an environment model is even more useful and necessary when the robot has to operate in that environment for longer amounts of time rather than just achieving a goal location. As the robot moves and builds the environment model, a correct estimate of robot location is necessary to ensure the spatial coherence of the environment model. Wrong or imprecise robot localization might corrupt the environment model constructed by the robot, which might lead to a failure in a task that the robot had to perform in the environment. Figure 4.2 from (82) shows the importance of localization for building environment models that are consistent with the real environment the robot is operating in. The figure (left) shows a robot only using odometry dead reckoning (*cf.* subsection 4.2.1) in order to localize itself and its localization estimate

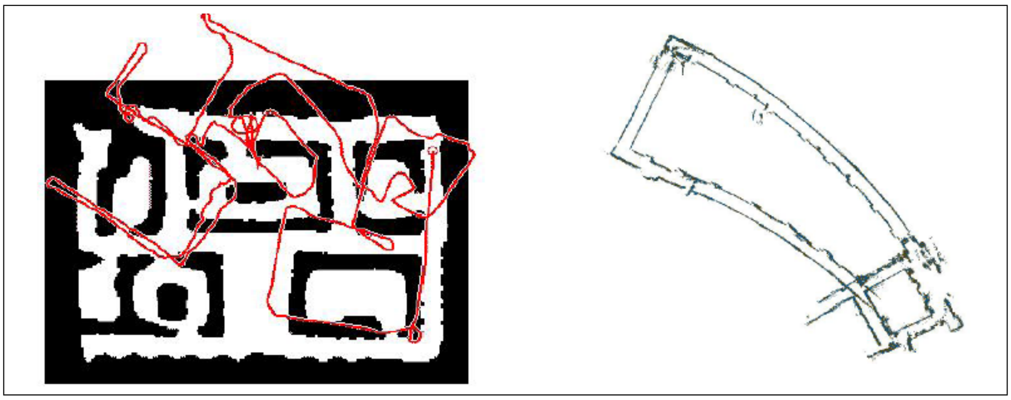

Figure 4.2: Effects of erroneous localization on environment model building, from (82). Left: Robot localization estimate acquired only using odometry drifts significantly with respect to the real environment. Right: The robot fails to recognize an already visited area (the data association problem) which induces incoherencies in the environment model being built.

drifts significantly with respect to its real movement in the environment. In such a situation if the robot has been mapping the environment using the perception data it acquired, the resulting model will be totally inconsistent with respect to the real environment. The figure (right) shows another instance of localization problem that can lead to inconsistent environment models. The figure shows a scenario where a robot uses odometry in order to localize itself, and as it approaches an already visited area it fails to recognize that this area has already been visited and thus inducing spatial incoherencies in the environment model being built.

4.2 Solutions to localization

Many solutions to the localization problem have been employed in robotics. Different solutions require different sensing sensing capabilities and computational power, and provide localization estimates with different characteristics. These solutions range from localization using only wheel odometry to the absolute localization anywhere on the globe using constellations of satellites revolving around earth. This section presents solutions to the localization problem in robotics including (i) dead reckoning, (ii) simultaneous localization and mapping (SLAM), and (iii) absolute localization using an a priori available map or a set of beacons.

4.2.1 Dead reckoning

Dead reckoning is the estimation of one's position by integrating the motion information over time. The motion information is acquired mostly through proprioceptive sensors like odometers, accelerometers and gyroscopes, but dead reckoning techniques using exteroceptive sensors like cameras also exist (e.g. (61)). Regardless of the type of sensors being used, dead reckoning in its raw form is an open loop process *i.e.* there is no feedback from any other sensors to correct the estimated position. Because of this lack of a feedback, the position errors grows constantly in dead reckoning. For a one dimensional case, the growth in position error is directly proportional to the square root of the motion variable but this error growth can be much larger when the robot can move arbitrarily in two dimensions because of the lever arm effects. In other words, in the two dimensional case a small drift in the angular measurement of the robot rotation leads to a large drift in its position estimate over time.

Most basic type of dead reckoning in robotics is done using odometry. The rotation of robot wheels is measured using shaft encoders and this rotation information is integrated over time to estimate the robot position over time. In case of a differentially driven robot with shaft encoders fitted on its wheels on both sides, the difference between the rotation measurements from the wheels from opposite sides of the robot indicates the change in robot orientation, and thus the robot orientation can also be estimated by dead reckoning. In practice, even if the shaft encoders are very precise, the position error grows quickly because of biases such as imprecise wheel radii and other factors such as of wheel slippage, uneven ground surface etc.

Inertial sensors such as accelerometers and gyroscopes are also used for dead reckoning in robot localization. Accelerometers are used to measure accelerations undergone by a body and gyroscopes are used to measure its rate of rotation. These two information can be integrated in time in order to localize a robot with respect to its initial position. The systems that use these sensors for dead reckoning are referred to as "inertial navigation systems". The advantage that inertial sensors offer in comparison to shaft encoders is that they do not suffer from wheel slippage. Nevertheless inertial-sensor-based dead reckoning also suffers from continuous increase in localization error because of measurement error integration over time. (11) and (62) present dead-reckoning robot localization using fiber optic gyroscope data fused with the wheel odometry data. Many inertial navigation units for high grade localization solutions are also available off the shelf for robotic purposes, one such example is the Applanix POS LV (4) which not only uses inertial sensors for localization purposes but also fuses the GPS data when it is available.

(75) presents dead-reckoning robot localization method using optical mouse sensors pairs. An optical mouse sensor can detect the motion in two dimensions so they use a pair of optical mouse sensors to sense robot motion in the two directions as well as the change in its orientation. The data from optical mouse sensors is integrated over time in order to perform dead reckoning. To make the technique more robust, they also present the use of four optical mouse sensors, where sensor readings are compared to detect and discard erroneous readings.

Another type of dead reckoning performed using cameras is the visual odometry. In such techniques features are extracted from images, which are then matched with features extracted from the succeeding images in order to estimate the difference in position and orientation of the camera. (61) present a visual odometry technique for monocular and stereo cameras that employs Harris corner feature extraction and matching from camera images. (26) present a visual odometry technique using a three-camera rig that employs extraction and matching of Förstner features from camera images.

Dead reckoning has also been applied for localization of walking robots, (69) is one such example.

4.2.2 Simultaneous localization and mapping

Simultaneous localization and mapping (SLAM) is the problem of a robot being autonomously able to build a map of an a priori unknown environment while simultaneously localizing itself in the environment.

While performing SLAM, a robot observes its environment and saves some sort of perceived information from the environment. This constitutes the “mapping” part of a SLAM process. While moving around, the robot keeps a track of it’s own location in with reference to the environment map being built, this constitutes “localization” part of the SLAM process.

A lot of work has been done on feature-based SLAM where some local features are extracted from the data perceived from the environment. These local features are often point features, but they can also be higher order features such as lines or planes. These

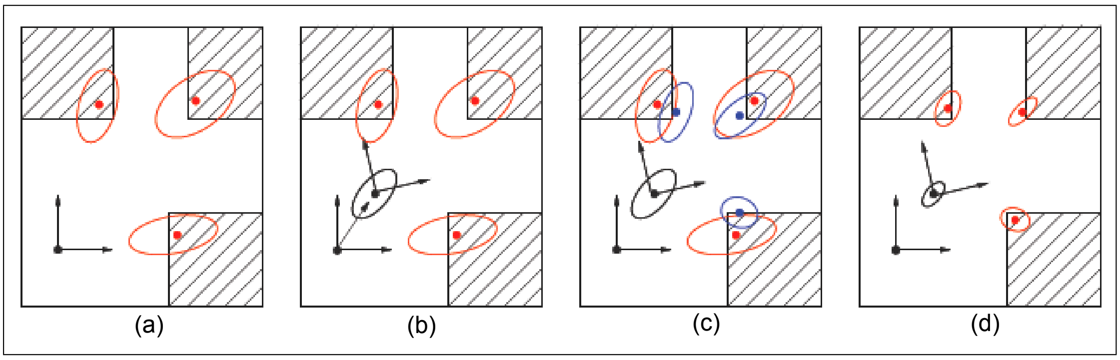

Figure 4.3: Feature based SLAM, from (39): (a) the robots observes its surroundings, detects three point features (shown in red) and adds them to its map as landmarks. (b) the robot moves and uncertainly in its position estimate grows. (c) the robot re- observes its surroundings and again detects three point features shown in blue. (d) the robot successfully associates the three newly observed to the landmarks in its map and updates its own position (and orientation) estimate and that of the landmarks.

features serve as the landmarks for building the map of the environment. Figure 4.3 from (39) depicts the concept of feature based SLAM. The robot observes some point features (shown in red) before it starts to move and adds them to its map of the environment as landmarks (figure-a). The robot then moves and the uncertainty in position estimate grows (figure-b). The robot then re-observes its surrounding and detects the three point features shown in blue (figure-c). The robot then successfully associates the newly observed features to the three landmarks in its map and and updates its own position (and orientation) estimate in the map and that of the landmarks (figure-d). While performing SLAM, robots update their own position estimate and that of the landmarks using different probabilistic filtering techniques. Extended Kalman filtering (EKF) is the most widely used probabilistic filtering technique to perform feature-based SLAM. Some implementations of SLAM systems employing EKF can be found in (40), (41) and (36). (56) introduced a Particle filtering based solution to the SLAM problem where the robot pose is estimated by a particle filter while EKF is used for estimating the landmark locations. Since the introduction of Particle filtering in SLAM it has also been widely used to perform feature-based SLAM. Some implementations of SLAM using Particle filtering can be found in (18) and (17). Finally and more recently, global optimization techniques have proved to solve the SLAM problem in a robust and efficacious manner.

A crucial aspect of the SLAM problem is the “loop closures”. A loop closure occurs when while performing SLAM, a robot visits a location that it already visited earlier and correctly detects this revisiting of the location (by correctly associating the observed features to the landmark in the map). By correctly detecting a loop closure, the robot can update any drift that has occurred in the estimates of its own position and that of the landmarks over time. Loop closures are very important in SLAM because a wrong loop-closure detection can lead to a completely corrupted map of the environment and thus incorrect localization estimates in the already corrupted map.

SLAM has also been performed without explicitly extracting local features from the environment perception data. Featureless SLAM techniques rely on the global appearance information in the environment. The seminal work in this regard comes from the computer vision community rather than the robotics community. (52) present a technique for view-based navigation using omnidirectional image sequences. The idea behind the work is that during a teaching run a robot memorizes the environment by saving a sequence of omnidirectional images. Then during an autonomous run newly acquired images are matched to the saved omnidirectional image sequence in order to determine robot pose. A more recent and very efficient technique for appearance-based SLAM is the FAB-MAP technique presented in (14). They use bag-of-words technique to develop a vocabulary which is then used to describe image data perceived at different locations in the environment and then Chow Liu trees are used to learn and match different locations in the environment. (3) also present a technique that uses bag-of-words technique for loop closure detection in SLAM.

4.2.3 Absolute localization

Read reckoning and SLAM presented in the last two subsections provide a localization estimate with respect to the starting position of the robot. Absolute localization, on the other hand, refers to the estimation of robot position and orientation with respect an already defined map/model or coordinate frame. Absolute localization techniques can further be divided into two categories *i.e.* beacon-based absolute localization, and map-based absolute localization. The two categories are explained below.

4.2.3.1 Beacon-based absolute localization

The principle lying behind the beacon-based localization techniques is "trilateration" where the 3D position of a point can be calculated using the distance of the point in question to four points with known positions in a reference frame.

An example of beacon-based localization are the global navigation satellite systems (GNSS) such as the American Global Positioning System (GPS) and the Russian GLONASS. Another GNSS that is in the phase of development is the European "Galileo". Such systems employ a number of satellites that revolve around earth in order to provide position estimates to an arbitrary location on earth. GPS is by far the most popular and widely used GNSS. GPS satellites constantly send out signals with a time stamp which can be received by a GPS receiver on earth. Using the GPS signal time stamps, the receiver can calculate its distance from the visible GPS satellites and thus estimate its location on earth represented in terms of latitude, longitude and elevation values.

GNSS have widely being used in robotics because of their global availability and low-cost sensing hardware. They however have a number of limitations and disadvantages. First of all the GNSS signals are only available in the outdoor environments so they cannot be be used for robot location in indoor environments. Even in the outdoor environments, and especially the urban outdoor environments, tall buildings and similar structures hinder the GNSS signals. Another problem in such environments is that the GNSS signals are reflected from building and other structures and are received by the GNSS receiver with a delay inducing error in the position estimation (multipath issues). Secondly any GNSS availability can not be guaranteed at all times because it can easily be jammed, or its availability might be ceased by the GNSS operator. Another limitation of GNSS is that GNSS signals cannot penetrate into water and thus cannot be used for underwater robotics.

Long range navigation (LORAN) is another example of beacon-based absolute localization. LORAN consists of radio signal transmitters around the world whose signals can be received by a receiver which then uses its distance to the transmitters in order to calculate its absolute position on earth. A signal from at least three LORAN transmitters is required in order to calculate the position of a LORAN receiver. LORAN has widely been used for maritime navigation.

(59) present a technique for beacon based localization for a fleet of robots where at all times any three of the robots within the fleet stay stationary and serve as beacons for the other robots. The other robots in the fleet move while localizing themselves to the three stationary robots. Then any other three robots from the fleet stop moving and serve as beacons while rest of the fleet is free to move within the visibility range of the current beacon robots. In this manner the whole fleet is free to move by following a leap-frogging strategy.

Another example of beacon-based localization is the localization in wireless Ethernet (WiFi) networks. In such systems WiFi transmitters serve and beacons and a receiver estimates its location is a given coordinate frame by analyzing its connection to the available WiFi transmitters and the corresponding signal strengths. Generally in such techniques, a signal-strength map of the available beacons (called the ”fingerprint” map) is built for a given environment during a learning phase. The built map is then used by robots to localize themselves in the environment by analyzing the signal strengths from different beacons at any given location in the environment. It is important here to note that the beacons measurements are not directly used to produce metric position estimates: indeed the distance estimated by the RSSI[1] of WiFi access points is extremely erroneous, especially in indoor environments where the signal propagates hardly along a straight line. The “fingerprint” techniques are actually similar to view-based localisation approaches (section 4.3). (30) present such a system where four WiFi transmitters are used as beacons in an office environment, and using a fingerprint map a robot is able to localize itself with an accuracy of up to 0.5m. Similar approaches have also been applied to GSM cellular networks: (63) present a technique that uses signal strength fingerprint maps of GSM cells, for absolute localization in multistory buildings with a median accuracy of 0.5m.

4.2.3.2 Map-based approaches

Map-based localization in robotics refers to the problem of localizing an autonomous robot in an environment whose map is a priori available to the robot. In map-based localization a robot senses its surroundings using its exteroceptive sensors, and then matches the perceived data to the available map in order to estimate its position in the map and the associated uncertainty.

---

[1] Received Signal Strength Indication

Simplest type of map-based localization techniques are the ones that rely on artificially placed markers or landmarks in the environment. Normally the artificial landmarks are arranged in the environment is such a way that the robot can infer a unique location in the environment by perceiving these landmarks. In other words these techniques are applicable to the environments with no perceptual aliasing. Perceptual aliasing occurs when two or more different locations in an environment lead to the same perception data because of limited sensor capabilities, and thus creating a confusion for the robot with no means to directly infer which one of the possible location is its correct position. One example of such a map-based localization techniques is presented in (8) where an autonomous underwater robot localizes itself using a grid of coded pattern. The pattern consists of groups of 8 circles that can either be black or grey. Five of these circles are used to encode position while the remaining three are used to detect orientation of the robot. Encoding the position with five dots each of which can take two colors leads to 32 possible unique patterns for the robot to infer its location on the grid. The robot perceives the grid using a downward looking came and infers its location and orientation on the grid. Figure 4.4 shows the robot and the grid coded pattern used by (8). (72) present a marker-based localization technique for indoor environments using coded patterns that can be hidden in the wallpapers or floor tiles in the indoor environments. The usage of artificial landmarks makes this set of localization techniques somewhat similar to the beacon-based absolute localization techniques, however they can still be classified as map-based techniques because the robot has to perceive its environment, match the perceived data with the a priori available map and infer its position.

Another category in the map-based localization techniques are the ones that enable robot localization in the environments that may have perceptual aliasing. In such techniques the robot perceives its environment, matches the perceived data to the available map, and probabilistically estimates its position in the environment. The initial position of the robot in the map might or might not be known to the robot and there are different solutions to solve the localization problem for the two cases. In case of a known initial position in the map, the robot starts to move in the environment and continuously updates its position hypothesis by observing the environment and incorporating the difference in its predicted position and the position inferred by observing the environment using a probabilistic filtering technique such as the Extended Kalman

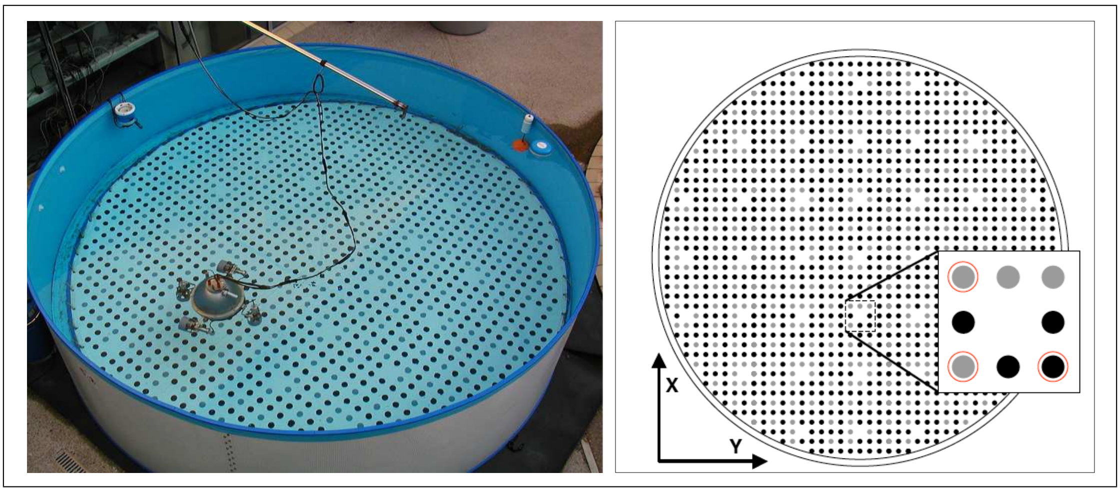

Figure 4.4: Localization of the autonomous underwater robot URIS using a pattern coded grid from (8). The robot and its environment (left), and the pattern coded grid (right) where three circles used to determine robot orientation are highlighted by red outlines in the zoomed-in view and the remaining five are used to encode a position on the grid.

filtering (EKF). Such techniques are called “single hypothesis tracking” techniques. A problem with single hypothesis tracking is that if because of some reason the robot loses the track of its correct position completely then its position is permanently lost and it cannot recover from the lost situation. Such situations can occur in case the uncertainty in the robot position grows large enough to hinder correct matching of perceived data to the map or if the robot is suddenly shifted to a new position in the environment without any indication of change in position to its proprioceptive sensors (referred to as the “kidnapped robot problem”). This problem is solved by the “multiple hypothesis tracking” techniques where a robot simultaneously keeps and updates multiple hypotheses of its current position in the map and different weights are assigned to each of these hypotheses depending on the likelihood of each one of them to be the correct one. Such techniques are implemented using probabilistic filtering techniques such as multi-hypothesis Kalman filtering or particle filtering.

(42) present a map-based localization implementation in urban outdoor environments using particle filtering. First a map of the environment is built using three Sick laser scanners fitted on a vehicle along an inertial navigation system (with GPS) to get the localization information for map building. The map that is generated is an overhead infrared image capturing the road texture such as lane markings and also the

vegetation near the roads. Using this map of the environment they are able to achieve a localization accuracy of 10cm when the GPS is used to assess vehicle velocity. The results are also provided for localization using lidar sensor and odometry data, and also only using the lidar data. For a 499m run using only the lidar data the localization error varies between 7cm to 35cm measured at five checkpoints along the 499m run.

(53) and (83) present map-based localization implementation for indoor environments using particle filters whereas (31) perform map-based localization in an indoor environment using multi-hypothesis Kalman filtering. (21) and (58) provide excellent reviews on map-based localization techniques in robotics.

In most of the map-based localization experimentation to date the maps of the environments are generated before the localization experimentation phase. Nevertheless another motivation for the research in map-based localization techniques is that often the maps of environments in which a robot is intended to operate are available, for example in the form of building architectural drawings for indoor environments and city maps and models for outdoor environments. The development of more robust and generic map- based localization techniques is surely a promising area for future research.

4.2.4 Choice of a localization method

The three main types of localization solutions presented above are complimentary to each other and each one has its own strengths and limitations. For example dead reckoning does not require sophisticated sensing hardware or computational power but the localization errors grows rapidly in dead reckoning. A robust SLAM systems can enable a robot to perform a task in an environment which was completely unknown to the robot a priori but a robust SLAM system requires rich sensing and high computational power. Beacon-based absolute localization methods enable a robot to accurately localize itself in a environment without requiring high computational power. But for beacon-based localization to work, the visibility of the beacons has to be guaranteed at all times during robot operation. For instance it might be feasible to take GPS availability for granted when developing a driverless taxi system for a city (civilian application) but it would not be feasible to let any autonomous vehicle developed to operate in the battle field to depend on GPS because the GPS availability cannot be guaranteed in a battle field.

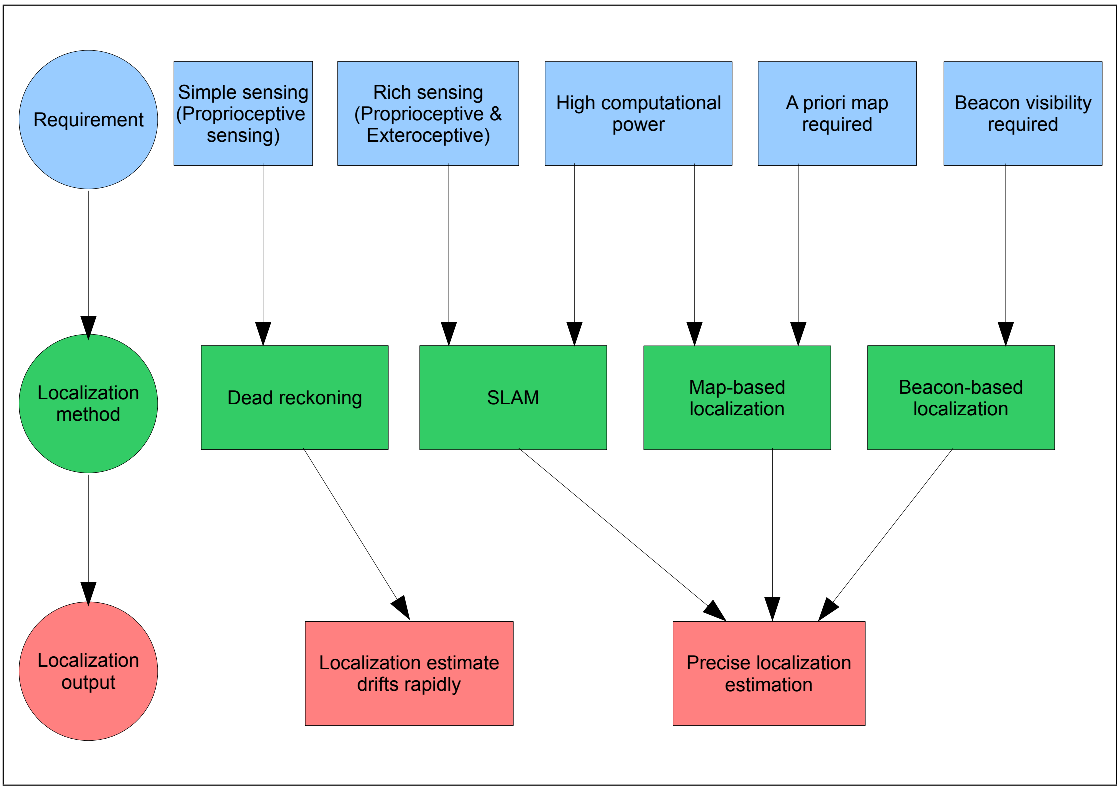

Figure 4.5: Comparison between different localization methods in terms of requirements (sensory, computational etc.) and achieved localization estimate.

In a nutshell a localization solution for a robot can be chosen by taking into account the two main factors *i.e.* (i) the intended application (the level of navigational autonomy required) and (ii) sensing and computational hardware at hand. Figure 4.5 shows a graphical representation of different localization methods in terms of sensory, computational and other requirements and the type of localization estimate achieved.

4.3 View-based localization

View-based localization refers to the localization techniques where a robot, while moving in an environment, observes its surroundings and builds a database of appearance information from its surroundings. As the robot continues to move, it continuously observes its surroundings and matches the current appearance information to that in the database. In this way the robot is able to localize itself in case it visits an area whose appearance information is already in its database.

View-based localization techniques find their roots in object classification where different objects can be identified as belonging to a certain class if it is similar to that class in appearance. (80) for instance, present a technique where appearance information of an object is encoded as histograms of color information in the object. These histograms are then used to identify a certain object in a given image.

View-based localization applies to two kinds of scenarios in robotics. First is the map-based localization. While being manually driven in an environment, a robot builds a database of appearance information of the environment. And later on the robot uses this database to localize itself in the environment during an autonomous run. Second is a scenario where a robot while autonomously moving around in an environment, incrementally builds the database of the appearance information of its surroundings and simultaneously uses this database to localize itself in the environment. In such a scenario, view-based localization is very closely related to the "loop closure detection" in SLAM. In other words when a robot localizes itself with respect to an already visited location in view-based localization framework, a loop closure detection occurs in terms of a SLAM framework.

Regardless of the type of two scenarios mentioned above, a view-based localization framework consists of following three main steps:

- extraction of the appearance information for the environment
- indexation of the appearance information in the form of a database
- devising an appropriate and efficient technique to match the current appearance information at a given time to the database in order to localize the robot

Extracting the appearance information from the environment is probably the most crucial choice among the three steps mentioned above because during the next two steps the form of the built database and the matching technique depend on the choice of appearance information representation chosen in the first step.

On a broader level, there are two types of options for the extraction and representation of appearance information for view-based localization. One is the extraction of "global attributes". Global attributes encode the overall appearance information of the scene at a given instant into a compact signature. This can be achieved for instance by applying a mathematical transformation to the scene information such as the Principal

Figure 4.6: Acquired image (left), greyscale image and selected sub-window (middle), and the corresponding Image Array (right), from (55).

Figure 4.7: SIFT features: size of the circle is proportional to scale of the feature, from (96).

Component Analysis (PCA), or by computing histograms of appearance information such as color and scene depth etc. Figure 4.6 presents an instance of global signatures called "Image Array" presented in (55). An image array is formed by taking a 640x480 image, converting it into greyscale, taking a 300x160 pixels sub-window at centre of the image and then adding the pixel values in each column of the sub-window. The second way of representing appearance information from the environment is by extracting "local attributes" from the environment perception data. This includes the extraction of point features such as Harris corners or SIFT (scale invariant feature transform (47)) etc. or higher order features such as lines or planes. For instance figure 4.7 from (96) shows the SIFT features extracted from an image in an urban outdoor environment.

4.3.1 Using global signatures

A seminal work in view-based localization is the work by (51) where they propose a technique for robot localization in indoor environments. First during a learning phase, a robot is manually driven through the corridors of a building. The robot is equipped with a forward-looking camera which acquires 512 x 512 pixel images. The images are resized to 32 x 32 pixel resolution in order to save computational power and memory. Depending on how different the images are from each other, some of these images are saved as global signatures representing the environment. Authors call this memorized signature sequence the “View-Sequenced Route Representation”. Later on during an autonomous phase, the robot moves in the environment and as it acquires new signatures, it matches with the previously memorized signatures using image correlation in order to localize the robot in the environment. In order to overcome the limitation of a small field of view of a forward looking camera in this work, the authors extend this technique to omnidirectional vision sensors in (52). In this work 512 x 512 pixel images acquired using an omnidirectional vision sensor are transformed to an image of 128 x 32 image after a cylindrical projection (*cf.* figure 4.8). These transformed images are used as global signatures for environment representation. During the autonomous run the newly acquired global signatures are compared with the memorized signatures using template matching in order to localize the robot in the environment. Omnidirectional vision sensors are indeed very attractive for view-based localization because of their 360° field of view, and therefore they have extensively been used for such techniques.

(50) present another technique for view-based localization using omnidirectional images. They use the omnidirectional images as global signatures. During a learning phase the signatures from the environment are memorized by the robot. In order to localize the robot, a newly acquired signature is compared with the signatures in the database be computing correlation between pixels taken from five circles of different radii in the signature images. The reason behind performing correlation at different radii is that in omnidirectional images the pixels closer to the centre of the image represent the areas closer to the robot whereas the pixels farther from the centre of the image represent the areas farther form the robot. Therefore the pixels from circles at shorter radii are expected to change more as the robot moves compared to the pixels at larger radii.

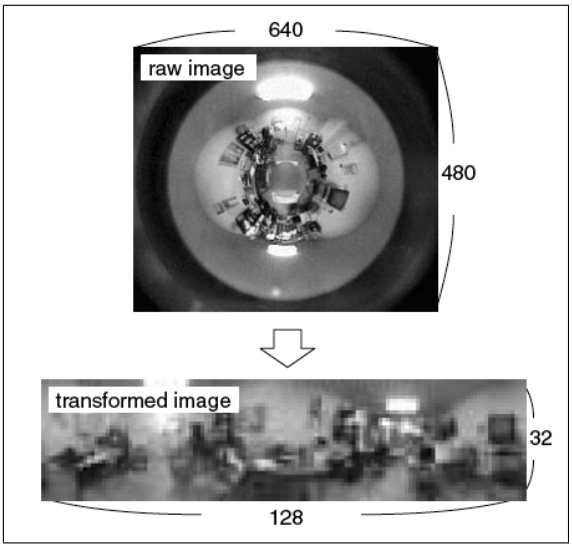

Figure 4.8: The omnidirectional-image-based global signatures from (52).

Instead of memorizing the raw omnidirectional images, one way of representing appearance information as a global signature is by applying Principal Component Analysis on the omnidirectional images. (93) and (32) present two such view-based localization techniques using omnidirectional sensors for indoor environments.

(24) present another method for extracting global signatures from omnidirectional images for view-based localization in outdoor environments. Five characteristics are empirically defined on the basis of Gaussian derivatives computed on the omnidirectional images. The five resulting images are split into three rings each. The rationale for splitting the images into rings, as mentioned earlier, is that small and large rings correspond to areas nearer and farther from the robot respectively and thus the appearance information in smaller rings change faster than in the larger rings. For each ring, a histogram is computed for the values of each of five defined Gaussian- derivative-based characteristics. This set of histograms computed for each omnidirectional image is used as a global signature to encode the appearance information. As the robot moves, the acquired global signatures are indexed during a database building phase. Later on during a recognition phase newly acquired signatures are compared with indexed signatures in order to localize the robot. Signatures are compared using different histogram distance measures including Histogram-intersection, Euclidean, Quadratic, Mahalanobis, and Haussler's distance and χ^2 statistic. Authors have found χ^2 statistic

to perform the best. (84) encode the appearance information from color panoramic images as global signatures by computing histograms of each of the three bands of HLS (hue, luminance, saturation) and RGB (red, green, blue) color spaces. And thus global signatures consisting of six histograms each are used for view-based localization.

4.3.2 Using local signatures

View based localization has also been performed by extracting local image features such as interest points. (96) present a technique that uses SIFT key-points for view-based localization. In a first phase, SIFT key-points are extracted from the images acquired in an environment and are saved in a database. Later on during the recognition phase, the SIFT key-points from a newly acquired image are matched with key-points in the database in order to find a localization estimate. (2) present a technique that uses a modified version of SIFT key-point extraction algorithm for local feature extraction in order to perform view-based localization. This modified descriptor does not comply to the inherent scale-invariance property of SIFT. Thus during the recognition phase this results in feature matches only when the robot is in the vicinity of a mapped location. (66) present a technique that uses descriptors for affine covariant regions extracted from omnidirectional images for view-based localization. Omnidirectional images are synthesized by rotating a camera with the help of a pan-tilt unit. Three types of affine covariant region detectors *i.e.* Harris-Affine, Hessian- Affine and MSER (Maximally Stable Extremal Regions) are used to extract feature regions from the omnidirectional images. Constellations of the extracted features are used as signatures to represent images. During the recognition phase a query signature is matched with map signatures in order to localize the robot. (10) present a technique that uses planar patches extracted form images to perform view-based localization. They use a Graphical Processing Unit (GPU) to implement their technique in order to achieve robot localization in real time.

Some techniques use the Bag-of-words (13) techniques to perform view-based localization. In such techniques an image or a location visited by the robot is described in terms of a set of "visual words". Visual words come from a "dictionary" of words which is constructed by clustering similar visual descriptors extracted from the images in the environment. In this way an acquired image can be represented by a set of words. As the robot moves, the newly acquired images are continuously transformed into the bag-of-words description and robots current location is compared with previously visited locations in terms of “visual words” description and thus view-based localization is performed. (3) and (14) present two such techniques.

4.4 Conclusion

The ability to reliably localize itself is imperative for autonomous operation of a mobile robot. The main solutions employed by autonomous robots in order to localize themselves have been reviewed in this chapter. They include dead reckoning, SLAM, map-based and beacon-based absolute localization and the concept of view-based localization. Different localization solutions have their own sensory and computational requirements and provide localization estimates with different properties and levels of accuracy. A robot that has a combination of localization solution at its disposal can indeed localize itself more reliably than using a single localization solution. This calls for the integration of different localization solutions for robust autonomous navigation. View-based localization, for instance, can provide a robot the means to localize itself in a given environment and at the same time assist a SLAM system to detect loop-closures. View-based localization using cameras has extensively been investigated in the literature but such techniques for lidars, and especially the novel multi-beam lidar sensors, have not been explored to the best of our knowledge. This is what the next chapter is about: it presents a view-based localization technique using multi-beam lidar data.

Chapter 5

View-based localization using 3D lidar

As explained in the previous chapter, being able to localize itself in an a priory unknown environment is an essential property to endow a mobile robot with autonomy. The introduction of novel 3D lidar devices, such as the Velodyne HDL-64E S2 (90), have changed the way mobile robots perceive their environment. A sample scan acquired from the Velodyne lidar device along with an aerial view of the environment has been shown figure 1.4: this figure shows the ability of the device to provide rich 3D data about the environment in a mere 200 ms acquisition time (the device is here set to spin at 5 Hz).

This chapter presents a method for qualitative view-based localization using 3D lidar data. "Qualitative" localization refers to the fact that the localization information provided by the technique is not expressed in metric coordinates, but with respect to the places already visited by the robot: the qualitative localisation here states to which previously visited place the robot is the closest. The technique is based on the indexing of global signatures extracted from 3D lidar data. The technique, on a broader scale, applies to the view-based localization and place recognition frameworks for an autonomous mobile robot. In a SLAM context, the presented technique can be used to detect loop-closures, *independently from the SLAM system*, and thus assisting it to perform loop-closures.

This chapter first presents the extraction of global signatures from 3D lidar data along with some measures of calculating similarity between the extracted signatures.

Then the application of these global signatures for loop-closure detection and view-based qualitative localization are presented followed by results on real lidar data.

5.1 Global signatures for 3D lidar data

5.1.1 Local vs. global signatures

As explained in section 4.3, a basic choice in a view-based localization framework is the choice of a method for extraction and representation of appearance information from the environment. Appearance information from the environment can be extracted and represented in two ways. One is the “global signatures” that encode overall scene information into a compact representation e.g. by computing histograms of the scene attributes such as scene depth and color, etc. The second way is to extract “local signatures” from the environment such as interest points, lines or planes, and represent the scene in terms of sets of extracted local signatures.

(92) present an extension of the classical 2D Harris corner detector to 3D for range data. They present the interest point detection results for synthetic 3D data as well as for 3D scans of miniature objects. Such 3D data have very low noise compared to the 3D data acquired by the Velodyne lidar, and therefore applying such interest point detectors to the Velodyne data is challenging. An interesting and innovative work about the extraction of interest points from lidar data has been presented in (44) and (45), where the Kanade- Tomasi corner detector on 2D and 3D lidar data is applied by projecting the data into 2D images. 2D lidar data is converted into an image by convolving it with a Gaussian kernel whose width is adjusted according to sensor range noise as well as positional uncertainty of each scan point arising from scan sampling/discretization. One advantage of this Gaussian smoothing is that at short ranges it smoothes the scanned surface which might otherwise look noisy as a result of range-measurement error. In order to rasterize 3D lidar data, the authors discretize the scan into a 2D grid (in horizontal plane). Then in each cell, the maximum and minimum height of present points is checked and their difference is calculated. An image is then drawn based on this height difference value for each grid cell, and a corner detection is applied to the image. The authors present results in both natural and indoor environments for 2D data and outdoor 3D data (one 360° Velodyne scan) and the method appears to perform well. (60) presents a comparison of line extraction

algorithms for 2D Lidar data. These algorithms include Split and merge, RANSAC and Hough-transform etc. But the implementation of these techniques on noisy 3D lidar data such as from the Velodyne is yet to be investigated to the best of our knowledge.

(15) present a method for automated extraction of planes and calculation of transformation parameters between 3D laser scans using the matching planar patches from the scans. The method uses region growing for plane extraction which is computationally very expensive and especially for high density 3D data captured at fast rates such as from the Velodyne. (95) present a plane extraction algorithm for 3D point cloud data. The algorithm discretizes the point cloud into rectangular blocks and then RANSAC based plane extraction is applied to each rectangular block. Planes in each rectangular block are merged by region growing. They also provide results on synthetic 3D data. One of the challenges in applying this method to the Velodyne data is that it is hard to define a size on which to discretize the environment for sensible plane extraction, and another challenge is the computational cost to process the large amount of Velodyne data.

Fast acquisition rate and noisy nature of the Velodyne lidar data makes it challenging and computationally demanding to extract robust local features from the lidar data and then use them for view-based localization. This in turn calls for the need of global signatures that are easy to be extracted and stored and are fast to be matched in order to perform view-based localization. We therefore propose a global signature that is based on local surface normal information extracted from lidar data. The local surface normal information can be extracted efficiently by exploiting the arrangement of the 64 laser beams in the Velodyne lidar device. The steps involved in the extraction of these local-surface-normal-based signatures from the Velodyne data and a method to compare and match the features is presented in detail in the following subsections.

5.1.2 Surface normal extraction

One way of extracting the local surface normals from a point clouds is to choose the nearest neighbors of a given point in the point cloud and then perform a plane fitting on this local neighborhood in order to find the surface normal at the point in question. This is a computationally expensive process. But as the 64 laser beams in the Velodyne device capture 3D data in a known sequence, it is possible to compute local surface normals efficiently by exploiting the laser beam arrangement. Both these methods of extracting local surface normals from Velodyne data are presented below. Surface normal extraction results for both the methods are also presented for the planar surface shown in figure 5.1. The surface is a section of the planar wall shown in the figure 3.3.

5.1.2.1 Using plane fitting in local neighborhood

(43) present a method for extraction of local surface normals from accumulated Velodyne point clouds. Point clouds captured by the Velodyne lidar are accumulated by using the robot pose information from IMU (inertial measurement unit). For a given point in the accumulated point cloud, its 20 nearest neighbors are selected from the whole point cloud and a plane fitting is performed on this local neighborhood to find the local surface normal at the point in question.

Figure 5.2 shows the surface normals extracted for the test planar surface using this method. For clarity the surface normals are shown for 25% of the 3D points in the point cloud chosen at random. For a given point, 50 of its nearest neighbors were detected in the point cloud and PCA (principal component analysis) based plane fitting was performed on the local neighborhood in order to compute the surface normal at the point in question. The figure shows that most of the surface normals are normal to the test planar surface and thus correct. The figure also shows some incorrect surface normal estimates especially along the edges of the test plane. This happens because at the edges of the planar surface the 50-point local neighborhood selected around a point is not guaranteed to form a planar patch that is parallel to the test planar surface.

5.1.2.2 By exploiting sensor geometry

(57) present a method of approximating local surface normals by exploiting the geometry of laser beam placement inside the Velodyne lidar device. For a given point four neighboring points are chosen that lie above and below it, and towards its right and left. Right and left neighbors are the points acquired by the same laser beam immediately before and after the point in question. Upper and lower neighbors are the nearest points acquired by laser beams which are immediately above and below (*i.e.* having higher and lower pitch angles respectively) the laser beam that acquired the point in question. Once the four neighbors have been chosen, the local surface normal is estimated by taking the cross products between the vectors formed by the given point and the four neighbors, and then geometrically averaging the resulting cross products.

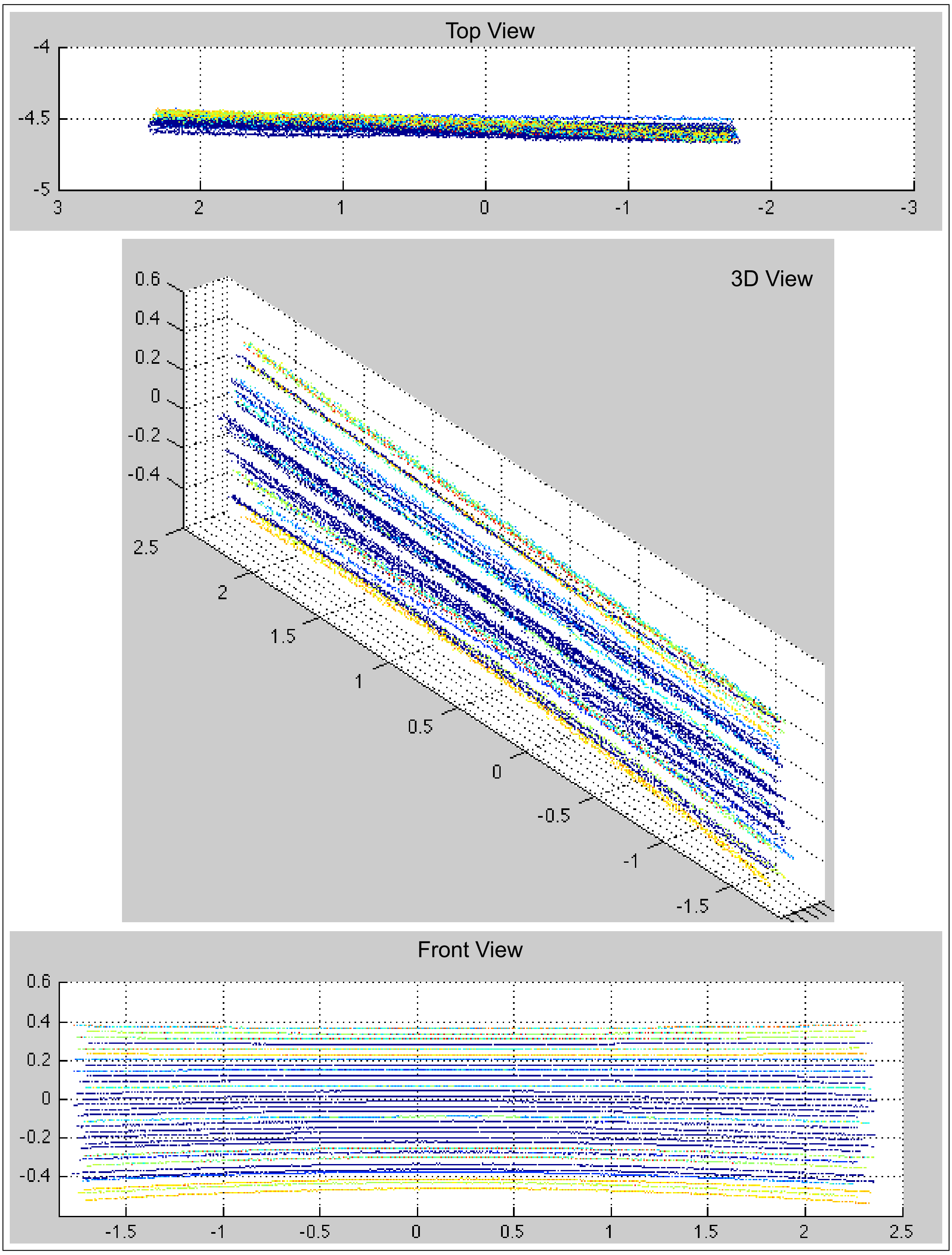

Figure 5.1: The test planar surface for surface normal extraction (units in meters).

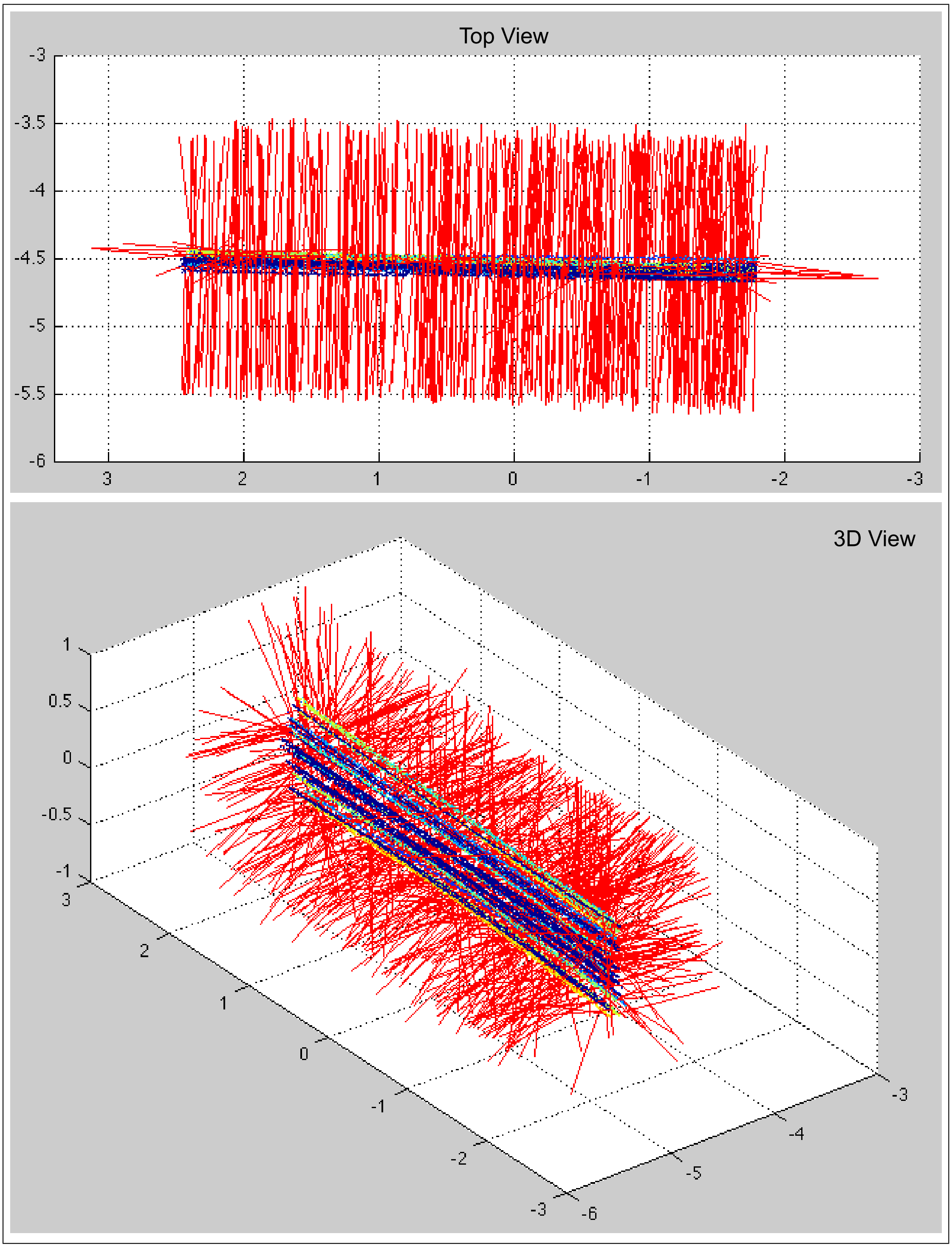

Figure 5.2: Surface normal extraction by plane fitting in local neighborhood (units in meters).

As mentioned in subsection 2.3.2, there is a horizontal misalignment between the laser groups lying on right and left-hand sides of the Velodyne lidar device (*cf.* figure 2.15 (top)). In terms of pitch angle, laser beams alternate between the laser groups on right and left side laser groups. In other words, for a given laser beam, the lasers beams two steps above and below it (in terms of pitch angle) belong to the laser-group located on same side (*i.e.* right or left) of the lidar head. This fact makes the points from laser beams two steps above and below a given point the best candidates for upper and lower neighbors rather than the points from laser beams immediately above and below the point in question (*cf.* figure 5.3 (top-left)). A naive choice for right and left neighbors of a given point would be the points acquired by the same laser beam immediately before and after the point in question (*cf.* figure 5.3 (top-left)). In practice however, a more appropriate choice for right and left neighbors are the points acquired five steps before and after the point in question (*cf.* figure 5.4 (top-left)). As the Velodyne data is noisy, selecting right and left neighbors five steps away from a given point guarantees that the vectors formed by the given point and its neighbors will lie parallel to the planar surface in which the given point lies.

The local surface normal estimation process using the four-point neighbor hood is shown in figure 5.5. Four vectors are drawn from the given point to its four neighbors. If all four of these vectors measure less than a threshold value, it means that the point in question is not lying on the edge of a surface and therefore a surface normal can be estimated at the point. Cross products are computed between right and upper, upper and left, left and lower, and lower and right vectors. The four cross product results are shown as blue lines in figure 5.5. The geometrical average of the four cross products (shown as the red line) is the estimated surface normal at the point in question. Figure 5.6 shows the surface normals extracted for the test planar surface of figure 5.1 using this method. In contrast to the the surface normal extraction using plane fitting, one advantage of the current method is that all the surface normals extracted for a surface consistently point away from the planar surface on the same side of the planar surface. Whereas in surface normal extraction using plane fitting, the extracted surface normal can point away from the planar surface on either side of the surface. This fact is apparent from the comparison of figures 5.2 and 5.6.

Figure 5.7 shows the surface normals extraction results for a single laser beam using the presented method in a semi-structured outdoor environment. The figure also shows

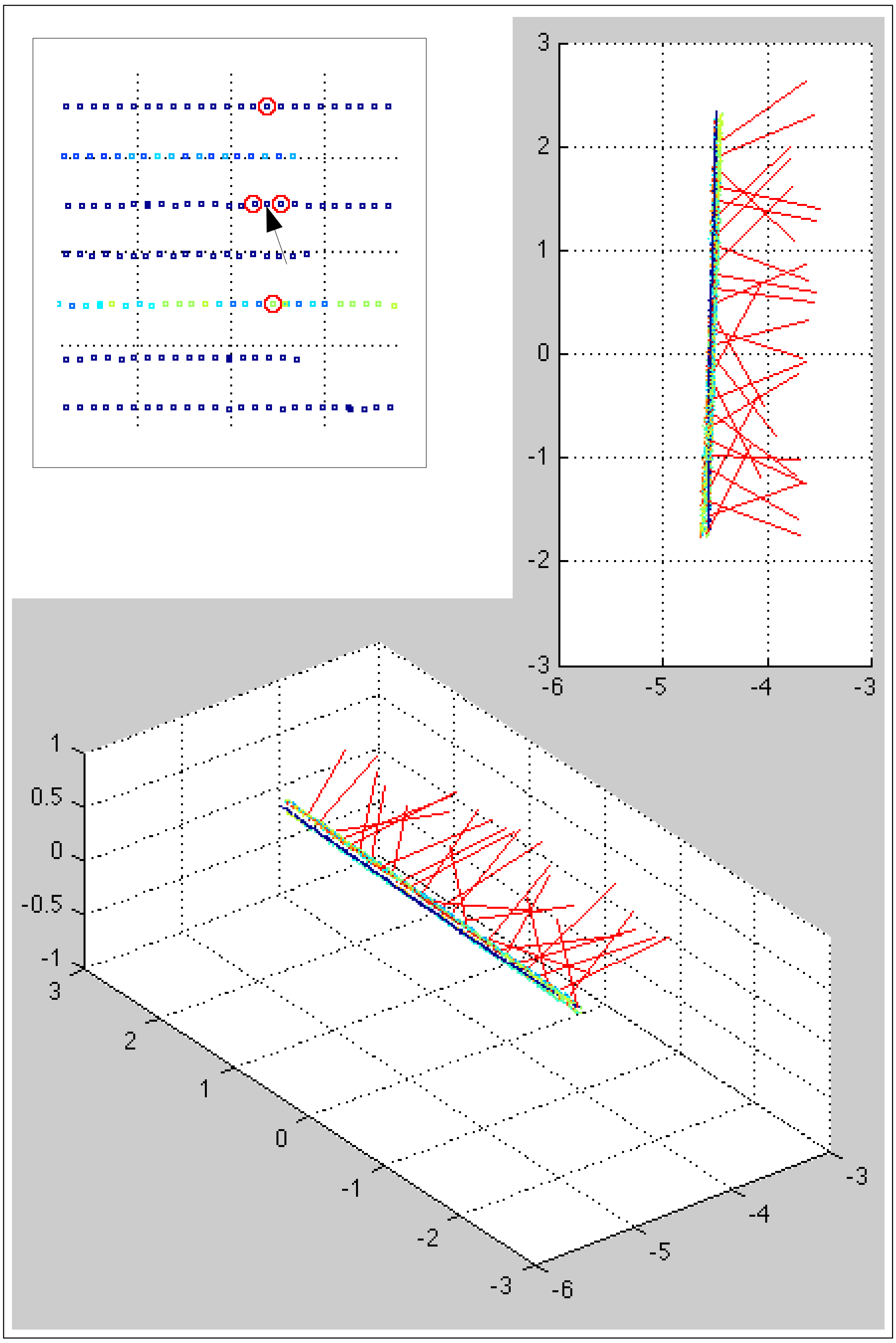

Figure 5.3: Choosing four neighbors of a given point (indicated by the black arrow), right and left neighbors are the points acquired by the same laser beam immediately before and after the point in question. Upper and lower neighbors are the nearest points acquired by the laser beams two steps above and below the laser beam that acquired the point in question (top-right). The resulting surface normals for a narrow planar patch (top-right and bottom).

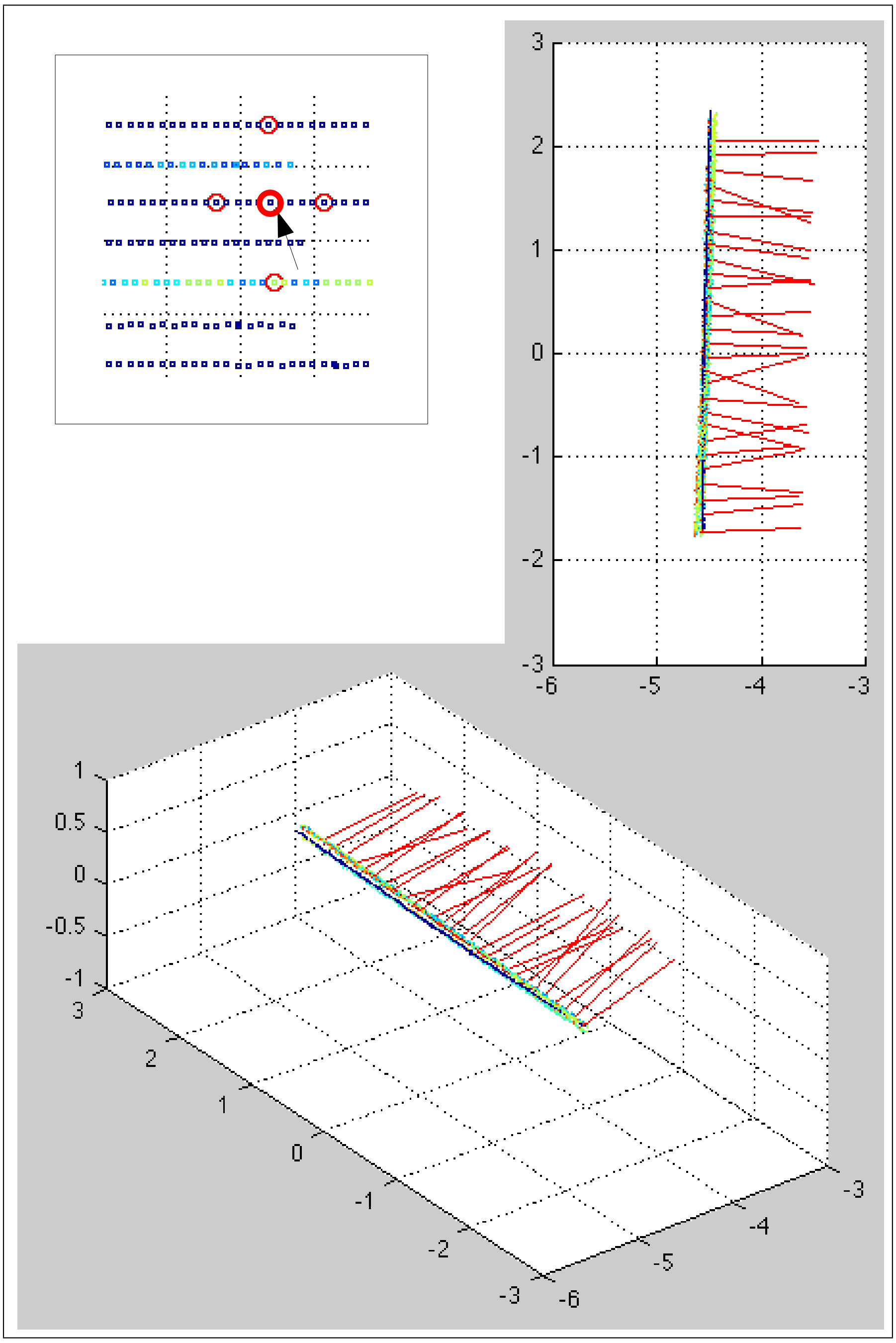

Figure 5.4: Choosing the points acquired five steps before and after a given point (indicated by the black arrow) as its right and left neighbors (top-left) and its effect on surface normal estimation for a narrow planar patch (top-right and bottom).

the aerial view of the environment.

The figures 5.3 and 5.4 show the effect of choosing the points only one step and five steps away from a given point as its right and left neighbors. The figures show the surface normal estimation results on a very narrow planar patch. It is clear that the surface normals estimated using right and left neighbors five steps away (figure 5.4) leads to the correct surface normal estimation instead of choosing right and left neighbors only one step away from the point in question (figure 5.3).

Among the two local surface normals extraction methods *i.e.* by plane fitting, and by exploiting the laser beam arrangement, the latter was chosen for the implementation of global signature presented later in this chapter because the method is not only more efficient but it also provides more consistent and accurate surface normal estimates.

5.1.3 Signature definition

Once the local surface normals have been extracted from the lidar data, a way to quantify the structure in the environment is by taking the dot product of normal vectors with vertical $\hat{z}$. When the robot is moving on a sufficiently flat terrain, the $z - axis$ of the $sensor - coordinateframe$ can be considered as the vertical. In case the terrain is not flat, the robot roll and pitch information can be used to transform the local surface normal vectors into a $global - coordinateframe$ and then their dot products can be computed with the $\hat{z}$ of the $global - coordinateframe$. The value of these dot products ranges between -1 and 1. Normal vectors that are parallel to the $x - y$ plane result in the dot product having low or nearly zero values whereas the normal vectors parallel to $z - axis$ result in dot products having values near -1 and 1.

For each 360° Velodyne scan, histograms of the above mentioned dot product results are computed after discretizing the range between -1 to 1 into 101 bins. These histograms serve as the global signatures for each Velodyne scan. Figure 5.8 shows three instances of such signatures extracted from outdoor lidar data. The figure also shows corresponding lidar scans and an aerial view of the the environment indicating where the corresponding scans were acquired. For the first two signatures (scans 1 and 1200) the higher values around histogram bins corresponding to the value “1” indicate the presence of ground are in the acquired scan. In other words, the surface normals for ground point in vertical direction and thus their dot product with $\hat{z}$ results in values around “1”. On the other hand the third signature in the figure (corresponding to

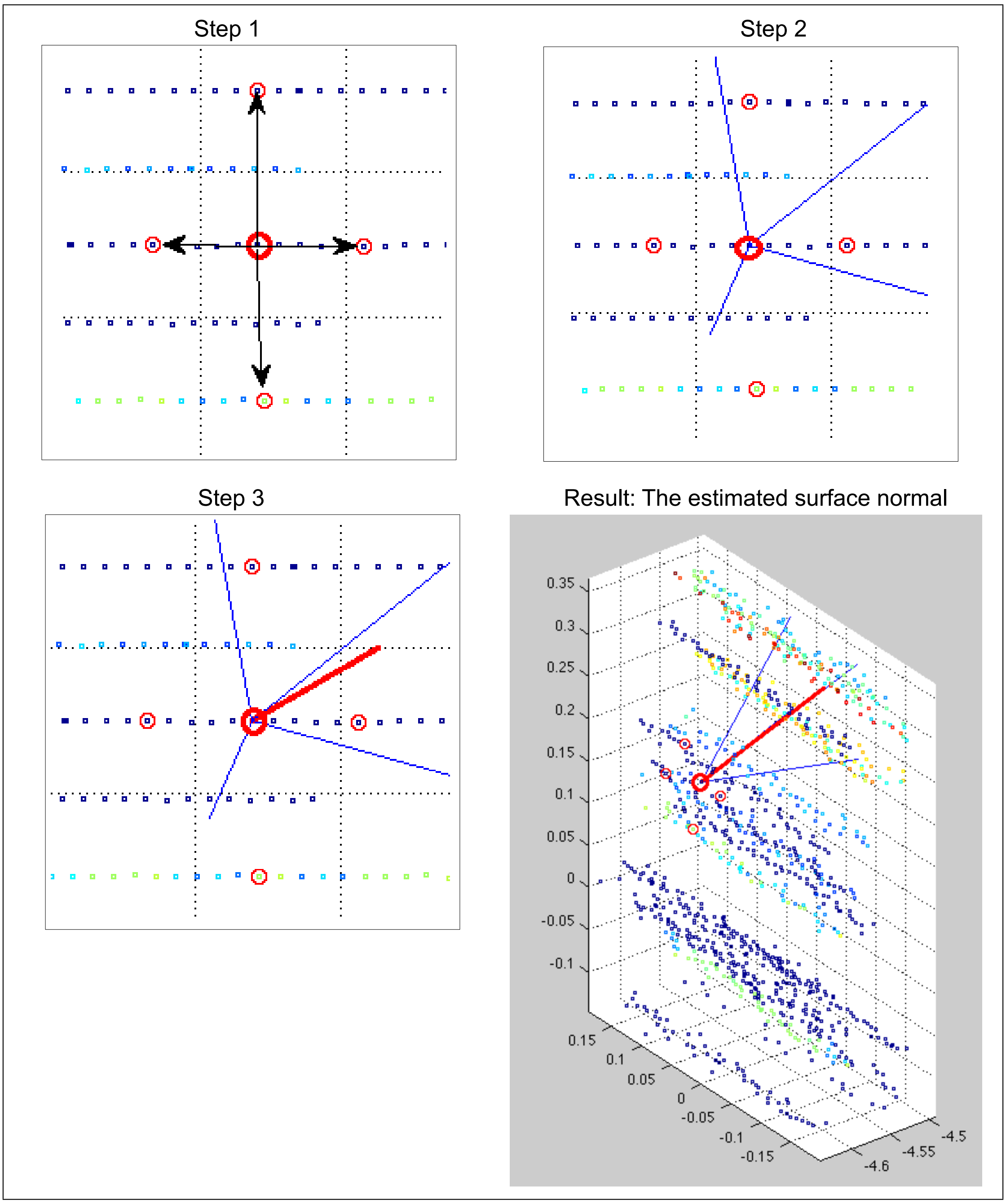

Figure 5.5: Surface normal estimation by exploiting Velodyne laser beam arrangement: Four neighbors and the corresponding vectors (top-left), the four cross products shown as blue lines (top-right), geometrical average of the four cross products shown as the red line is the estimated surface normal (bottom-left), and the result shown in 3D (bottom-right).

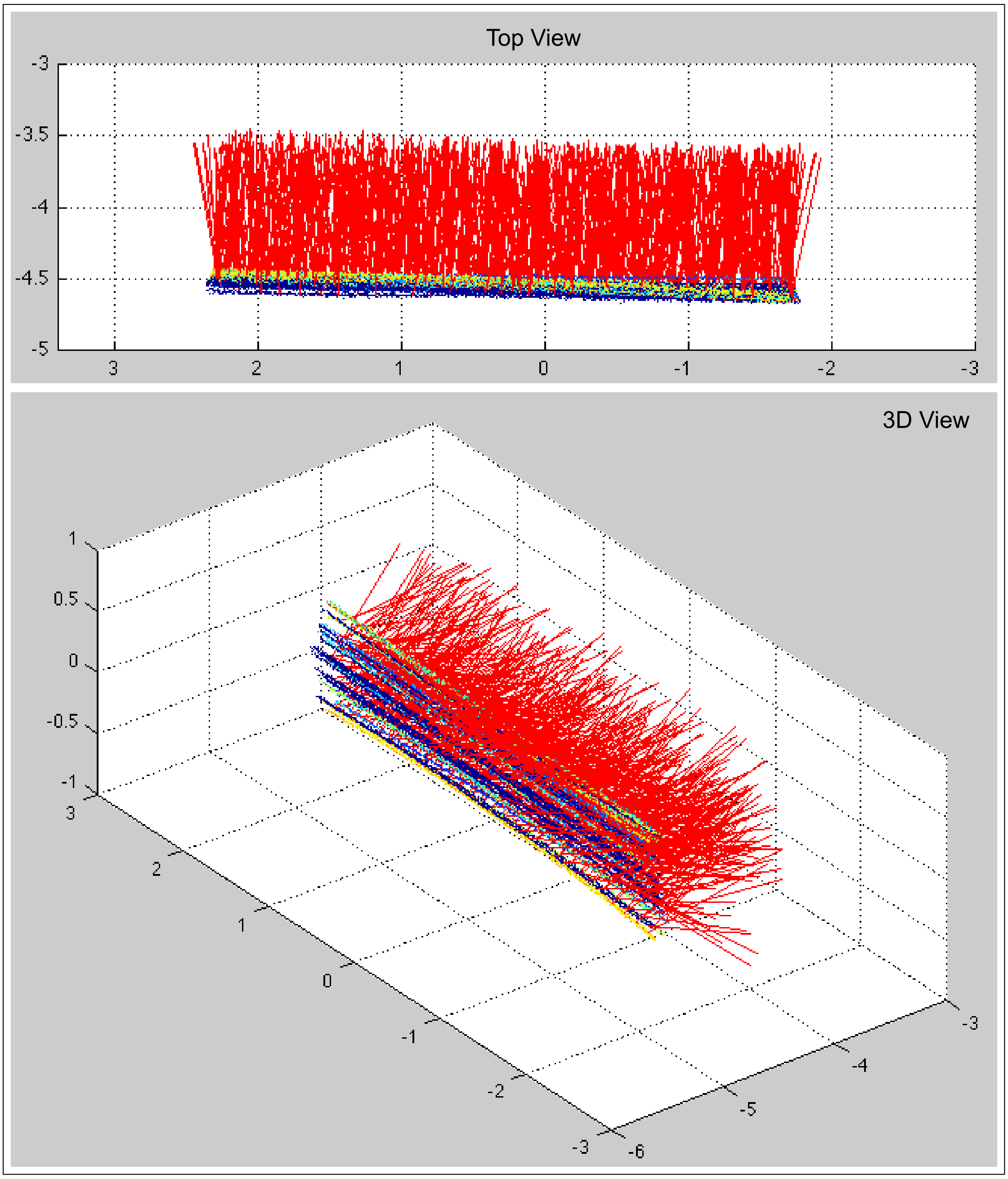

Figure 5.6: Surface normals extracted for the test planar surface of figure 5.1 by exploiting the laser beams arrangement in the Velodyne device.

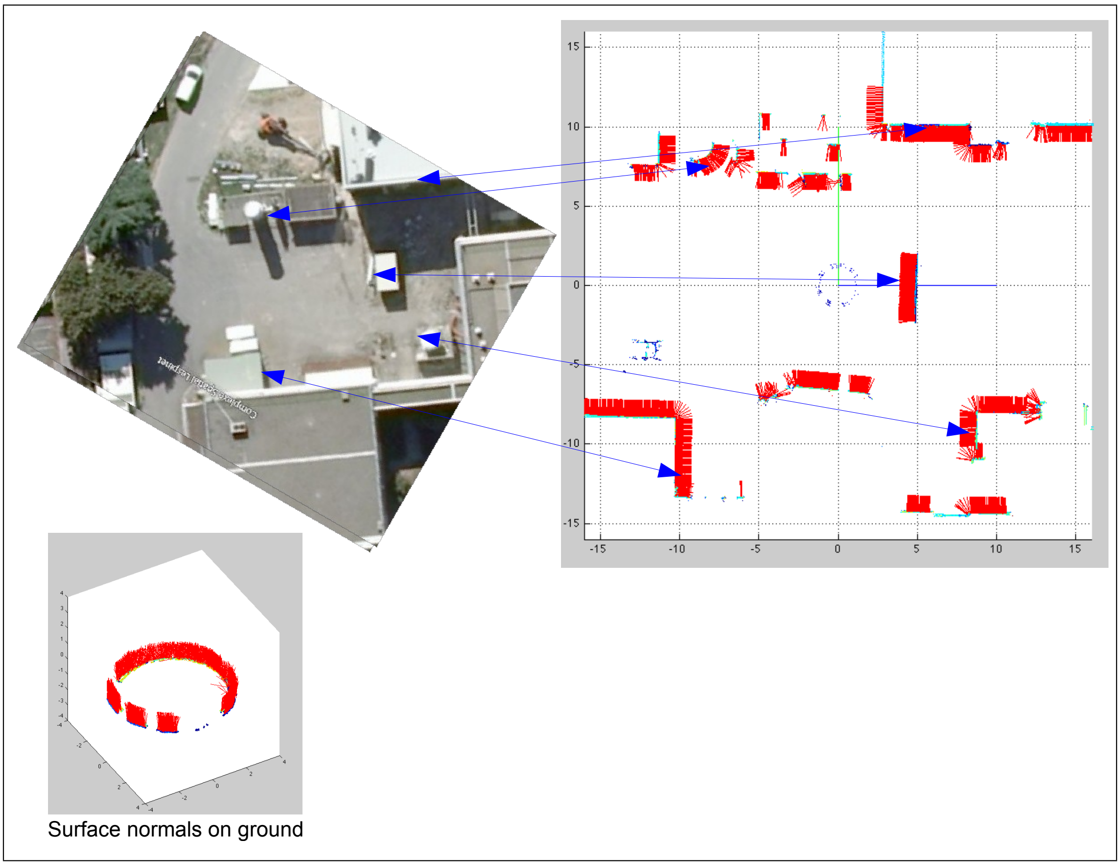

Figure 5.7: Surface normals (shown in red) for one laser beam along with the aerial image of the environment. Surface normals on ground for one laser beam (bottom-left).

scan 1320) has significantly lower values around the bins corresponding to value “1” compared to the other two signatures. This is because of the fact that scan 1320 corresponds to a closed space and therefore the ground area in this scan is smaller compared to the other two scans shown in the figure.

When the Velodyne device is set to run at 5 Hz, a raw 360° scan has a size of 548,625 bytes. One histogram signatures consists of 101 double floats and thus sizes 808 bytes. This gives a data reduction ratio of 679:1 for a Velodyne scan. For extraction of histogram signatures only the 3D points acquired within the depth range of 3m to 50m are used. The reason for discarding points less than 3m away from the sensor is that practically speaking most of these points are returns from ground when the robot is in an outdoor setting. The reason for discarding points beyond 50m range is that the working range of the sensor for low reflectivity objects is 50m as mentioned in the data sheet.

5.1.4 Comparing the signatures

An important and rather obligatory requirement after the definition of any kind of signature that encodes the appearance (or any other kind of) information is to devise a way to quantitatively compare the signatures. As the signatures proposed in the previous subsection are histogram based, a logical method to determine similarity/dissimilarity between the signatures is by calculating the “histogram distance” between signatures.

(84) and (24) report the use of different histogram measures to compare their histogram signatures (extracted from omnidirectional image data) in order to perform view-based localization. (84) have found “Jeffrey divergence” and χ^2 measures to perform best for their signatures while (24) have also found χ^2 measure to perform the best. (9) gives an overview of different distance and similarity measures between two histograms.

In our experimentation, among many possible candidates for histogram distance measurement (including χ^2, Euclidean, Sørensen, Lorentzian, Cosine, Fidelity and Jeffreys distances etc.), the χ^2 and Sørensen distance measures were found to be giving consistent distance values for different similar and dissimilar histograms. The two his-

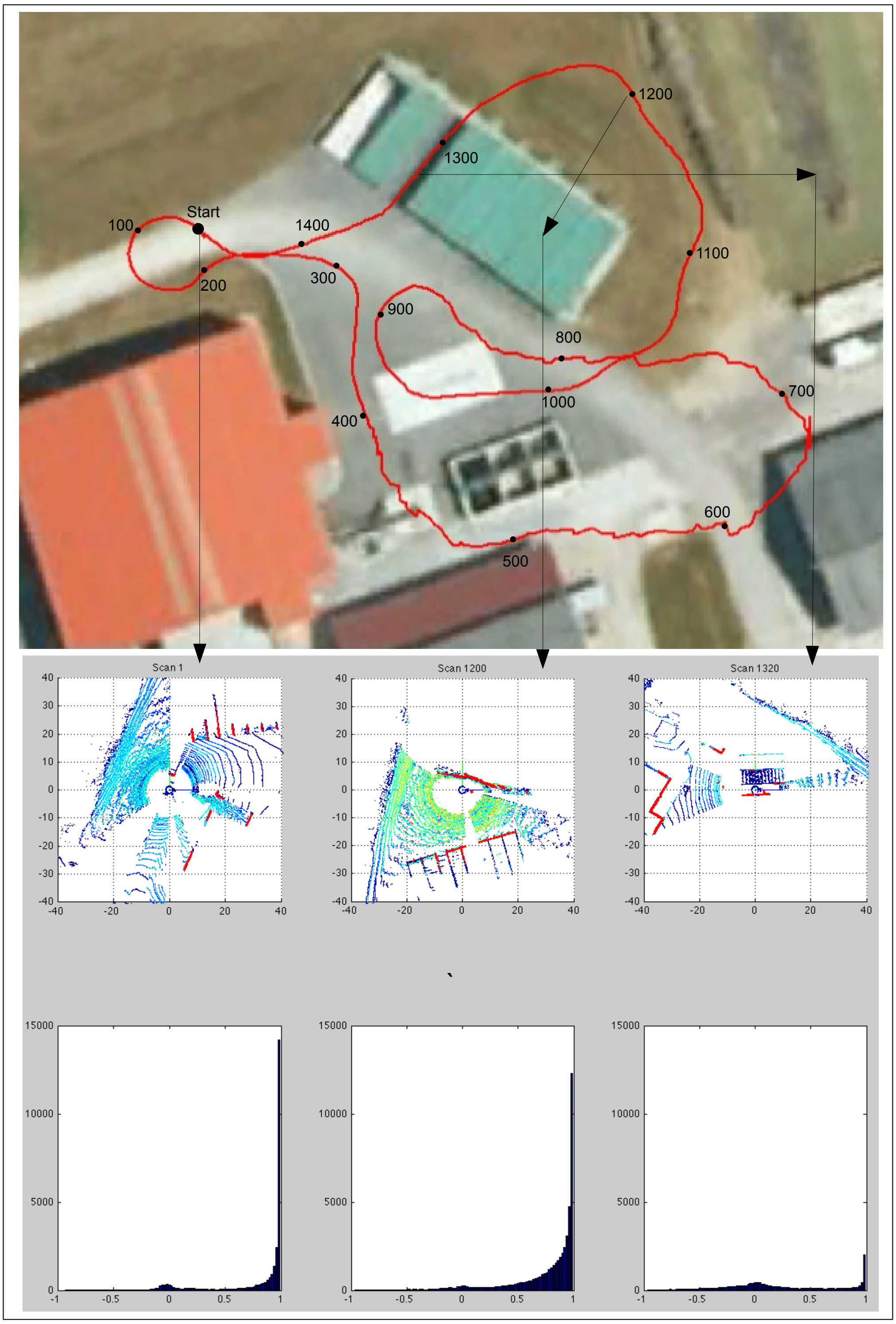

Figure 5.8: Three histogram signatures (bottom) along with corresponding lidar scans (middle) and an aerial view of the environment where the scans were acquired (top).

Table 5.1: Histogram distances between the three example signatures shown in the figure 5.8.

Scans	χ^2 distance	Sørensen distance
1 and 1200	14510.81	0.351
1 and 1320	17107.70	0.579
1200 and 1320	32038.13	0.654

tograms measures are defined as follows:

$$D_{\chi^2} = \sum_{i=1}^{d}((P_i - Q_i)^2/(P_i + Q_i + 1)) \tag{5.1}$$

$$D_{S\o rensen} = \sum_{i=1}^{d}|P_i - Q_i| / \sum_{i=1}^{d}(P_i + Q_i) \tag{5.2}$$

where P and Q are the two histograms and d is the number of histogram bins.

Table 5.1 shows the histogram distances between the three example signatures shown in figure 5.8. It can be seen that the histogram distances between scan-1320 signature and the two others (1 and 1200) is greater than the distance between 1 and 1200 themselves. This is intuitively correct because the signature for the scan 1320 is very different compared to the other two especially at histogram bins around the value "1" (*i.e.* pertaining to the ground surface normals).

Figure 5.9 shows the evolution of the two (χ^2 and Sørensen) histogram distance measures for the robot trajectory shown in figure 5.8. The figure shows that the two histogram distance measures show slightly different characteristics (regions highlighted by green circles). Therefore we suggest the use of both histogram distance measures combined in order to compare any two signatures.

5.1.5 Key-signature selection

Since the acquisition rate of the lidar device at hand is fast[1], the difference between two consecutive scans acquired by the device is slight. In turn, the corresponding signatures extracted from the scans are very similar, and thus comparing a signature to each and

---

[1] In our experiments, the Velodyne spins at 5 Hz.

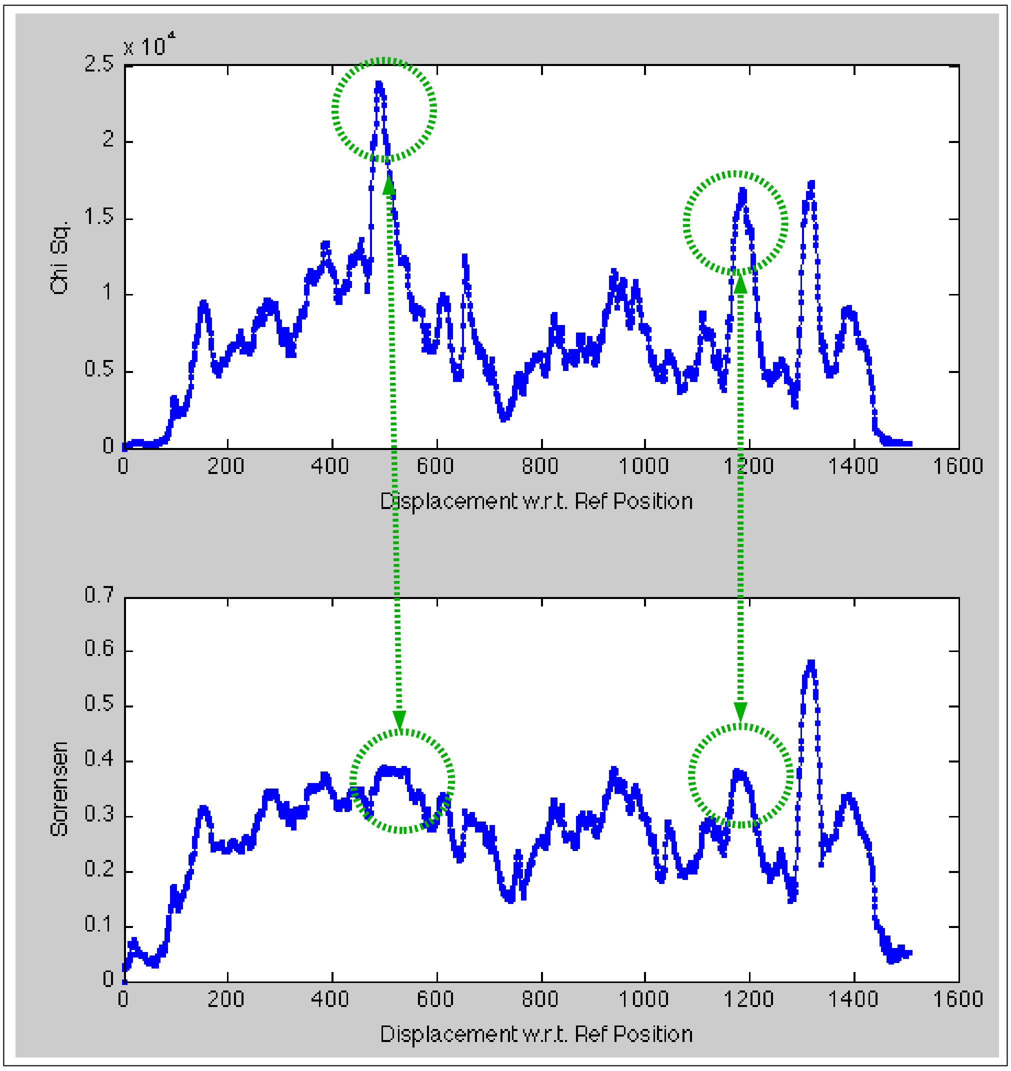

Figure 5.9: Evolution of χ^2 and Sørensen histogram distance measures. Scan no. 1 from the robot trajectory shown in figure 5.8 is taken as the reference scan and distance evolution is shown for proceeding 1504 scans acquired along the trajectory.

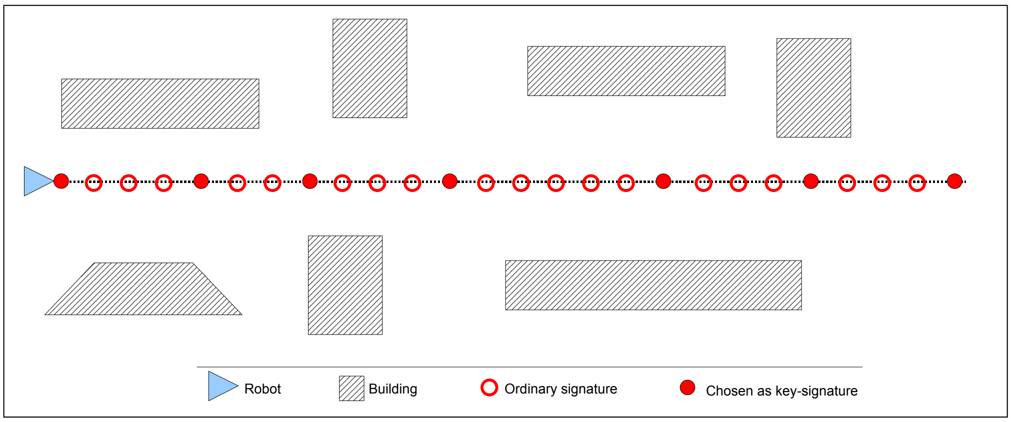

Figure 5.10: Key-signature selection: as the robot moves, some signatures are chosen as key-signatures depending on how different they are from the last chosen key-signature.

every previously stored signature is not strictly required in order to perform loop-closure detection or place recognition. For instance, when the Velodyne HDL 64E-S2 lidar device is set to run at 5 Hz, if the robot runs at a speed of 1m/sec, the distance at with the two consecutive scans are acquired is only 20 cm. We therefore use the concept of "key-signature selection". When the robot starts to move, the signature extracted from the first acquired scan is stored as the first key-signature. As the robot continues and new scans are acquired, the χ^2 distance between each newly extracted signature and the last key-signature is computed. A new signature is stored as a key-signature if the distance between the last key-signature and the current one is more than an empirical threshold T_{key}, and the process continues. The process is of key-signature selection is shown figure 5.10 where red circles represent an extracted signature and filled circles are the signatures chosen as key-signatures.

5.2 Loop closure detection

The small-sized global signatures for 3D lidar data defined in the previous section can be used for fast and efficient "place recognition" or "loop-closure detection" during autonomous robot navigation. "Fast and efficient" owes to the fact that the global signatures extracted from the 3D lidar data are very small is size compared to the raw lidar scans (679:1 data reduction ratio, *cf.* subsection 5.1.3) and the quick signature comparison using histogram distance measures. This section presents a method of loop-closure detection using the defined signatures along with results on Velodyne lidar data.

5.2.1 Methodology

As the robot moves in an environment and 3D lidar scans are acquired, their corresponding extracted global signatures are stored. Some of these signatures are marked as the key-signatures as explained in subsection 5.1.5.

At any time t, lets denote the signature extracted from the most recently acquired 3D scan by S_{curr} and each of the previously stored key-signatures by S_{k-i} where i ranges from 1 to n, n being the total number of key-signatures stored up to time t. The χ^2 and Sørensen distances are computed between the current S_{curr} and S_{k-i} for i ranging from 1 to $n - m$. m here is an arbitrary number which limits the histogram matching to the scans that were acquired sufficiently earlier than the current scan. In our experimentation we have set the value of m to 15. This value of m is just a matter of choice and smaller values of m might result in loop closure detection for the locations that the robot recently visited and therefore might not be of interest – a better solution would be to rely on the motion estimates provided by other localisation sensors or technique. For instance if the value of m is set to 2 and the robot is moving in a straight line at a slow speed, the loop closure detection algorithm would signal the detection of a loop closure just because the current histogram would match the key-histograms very recently acquired, whereas in reality this is not a real loop closure because the robot has not even left this area.

During the histogram matching step, if both χ^2 and Sørensen distances between histograms S_{curr} and S_{k-i} are smaller than the two threshold values T_{χ^2} and $T_{Sør}$ a "loop closure candidate" is registered. If this occurs at multiple key-locations i, all of them are registered as loop closure candidates. The location i which has the minimum Sørensen distance among all the loop closure candidates registered is taken as the loop closure detection location for the current scan.

Figure 5.11 shows the above explained loop- closure detection process. As the robot moves, some of the extracted signatures are chosen as key-signatures (shown as filled red circles). At the current time instant, the currently extracted signature S_{curr} (shown as the filled blue circle) is compared with previously stored key-signatures S_{k-i} for i ranging from 1 to $n-m$. In this figure n *i.e.* the total number of key-signatures is 8 and

m is set to 2. Therefore the S_{curr} is compared with key-signatures S_{k-1} to S_{k-6} and a match is detected for S_{k-1} and hence a loop closure is signaled between the current robot position and position corresponding to S_{k-1}.

5.2.2 Results and discussion

The loop closure detection technique was implemented on a data set gathered by our Velodyne sensor fitted on the robot Robufast shown in figure 5.12. The data set consists of 1505 360° lidar scans which correspond to nearly five minutes of robot motion and data acquisition. The Velodyne lidar device was set to run at 5 Hz during the data acquisition and the length of robot trajectory is approximately 375m. The robot was also fitted with centimeter accuracy D-GPS sensor in order to acquire the ground truth localization information for the experiment.

Fig. 5.13 shows the an aerial view of the experiment environment along with the robot trajectory (top) and the detected loop closures (bottom) shown as red lines between the corresponding robot positions. The signature comparison threshold values T_{χ^2} and $T_{S\o r}$ were set to 434 and 0.0391 respectively. The threshold value for key-signature selection T_{key} was set to 380 which lead to the selection of 255 (out of the total 1505) signatures as the key-signatures. Key-signatures are shown as the filled blue circles along the robot trajectory in the figure.

The robot starts to move and around the positions 893 its detects a loop closure with position 312. This is in fact the first time the robot comes near an already visited location. As the robot moves further, it detects more loop closures in this area. A zoomed-in view of this area is shown in figure 5.14 (top-left). The figure (middle and bottom) also shows the 3D scans and the corresponding signatures for one of the loop closure instances in this area (between the positions 918 and 332). The loop closure detection is shown as the black line in the figure (top-left).

As the robot continues to move, it detects a loop closure at the position 1056 with the already visited position 754. This loop closure detection along the corresponding 3D scans and signatures is shown figure 5.15. The two 3D scans in the figure (bottom) show that that scans are very similar in appearance with an approximately 190° shift in orientation as the robot was heading in opposite directions when the scans 1056 and 754 were acquired.

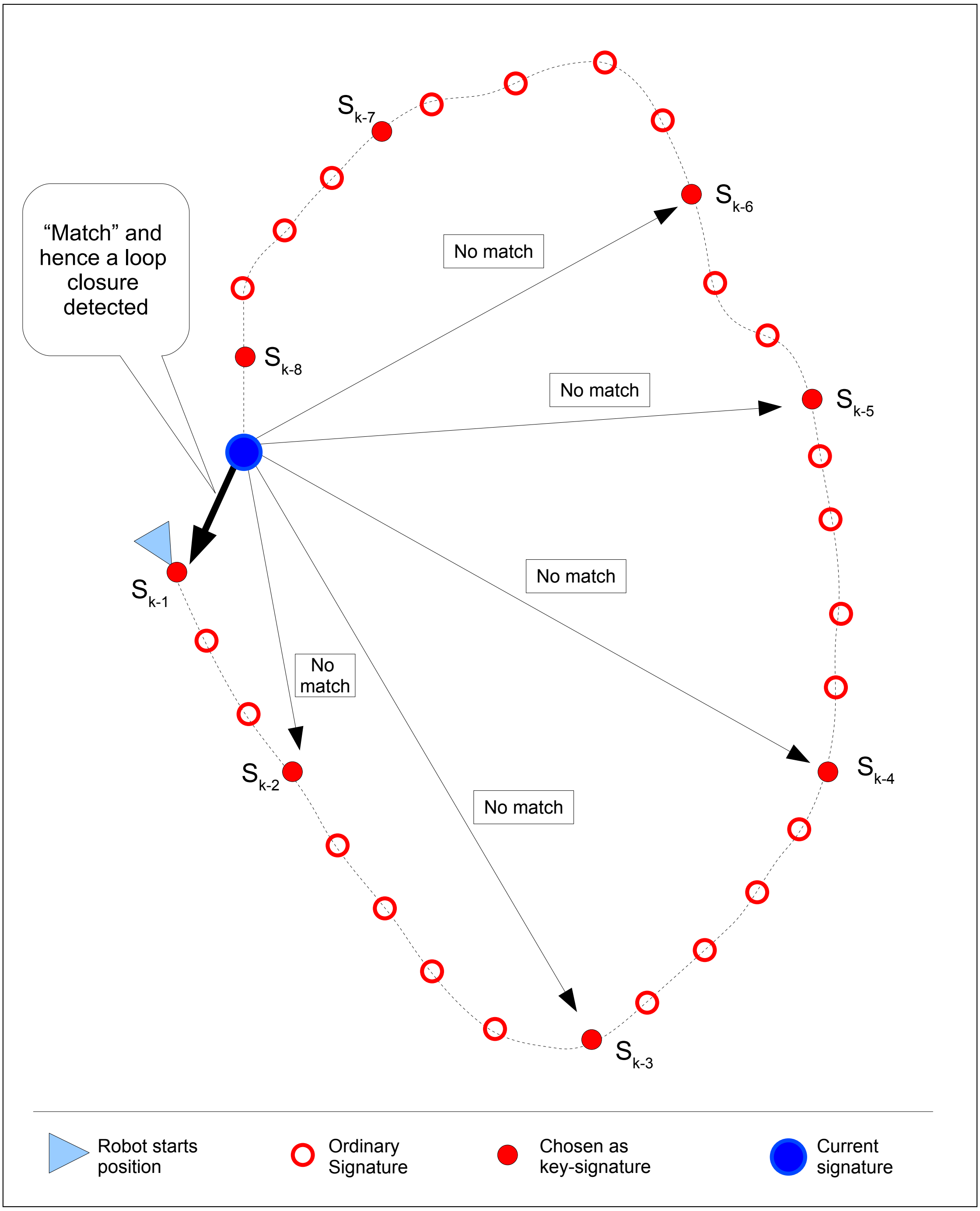

Figure 5.11: The loop closure detection process.

Figure 5.12: The robot Robufast equipped with the Velodyne lidar device and other sensors.

The robot continues to move and as it approaches position 1375, it detects a loop closure with location 317. As the robot moves it again registers several loop closures in this area. A zoomed-in view of the loop-closure detections in this area is figure 5.16 (top-left). The figure (middle and bottom) shows the 3D scans and the corresponding signatures for one of the loop closure instances in this area (between the positions 1380 and 281). This loop closure detection is shown as the black line in the figure (top-left).

As the robot continues to move and approaches the position where it started, it again registers several loop closure detections. Figure 5.17 (top-left) shows a zoomed-in view of these loop closure detections at the start and finish positions of the robot trajectory. The figure also shows the signatures (middle) and 3D scans (bottom) for one of the loop-closure detections at this location. The two scans are very similar in appearance as they essentially come from the same location in the environment.

Figure 5.18 (top-left) shows a zoomed-in view of the area in the experiment environment where false positive loop closures were detected between the locations around 950 and 544. The figure also shows the extracted signatures (middle) and 3D scans (bottom) for one of these loop- closure detections *i.e.* between the positions 952 and

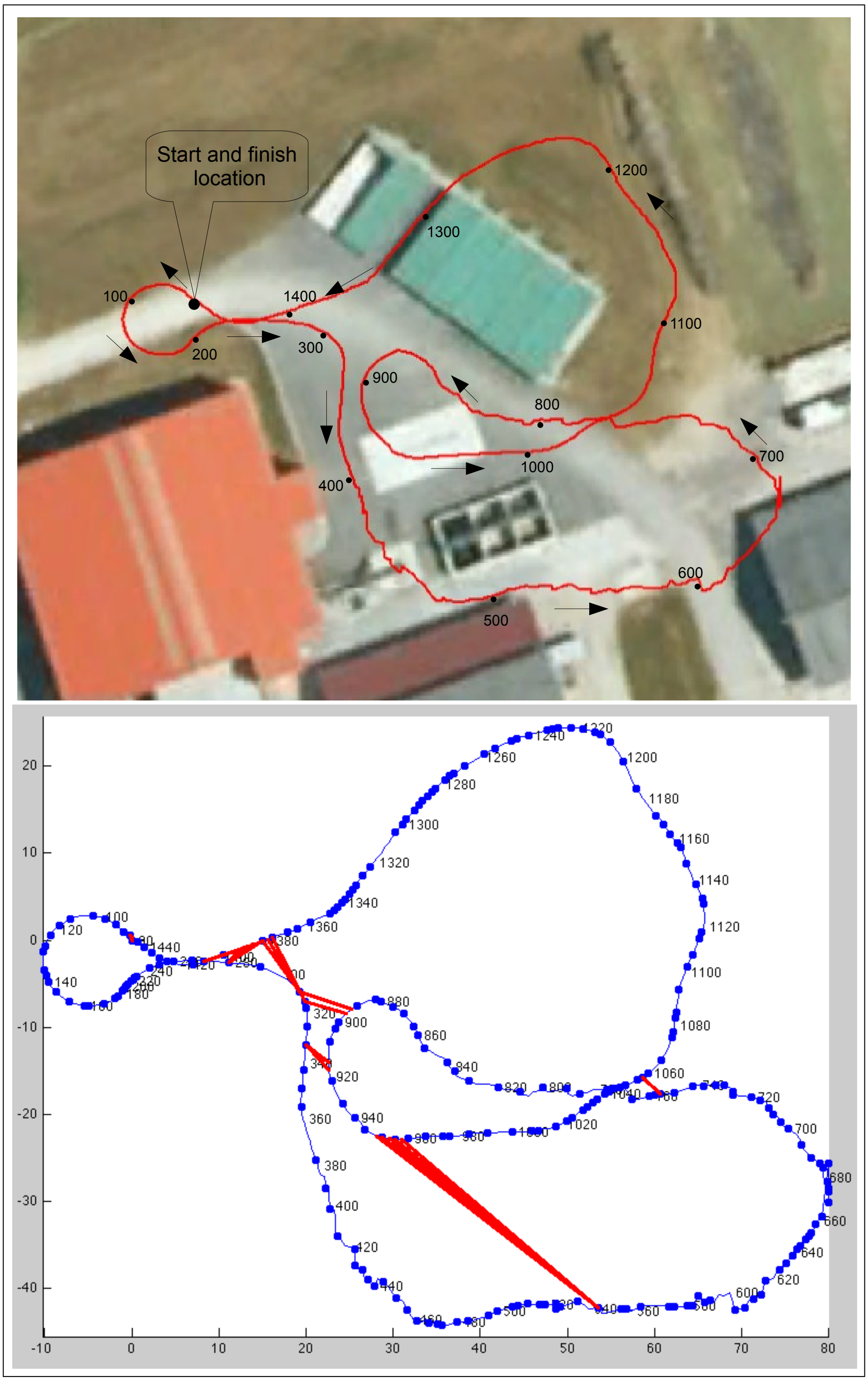

Figure 5.13: Aerial view of the environment along with the robot trajectory (top), and the detected loop closures shown as red lines (bottom), key-signatures are represented as filled blue circles along the robot trajectory.

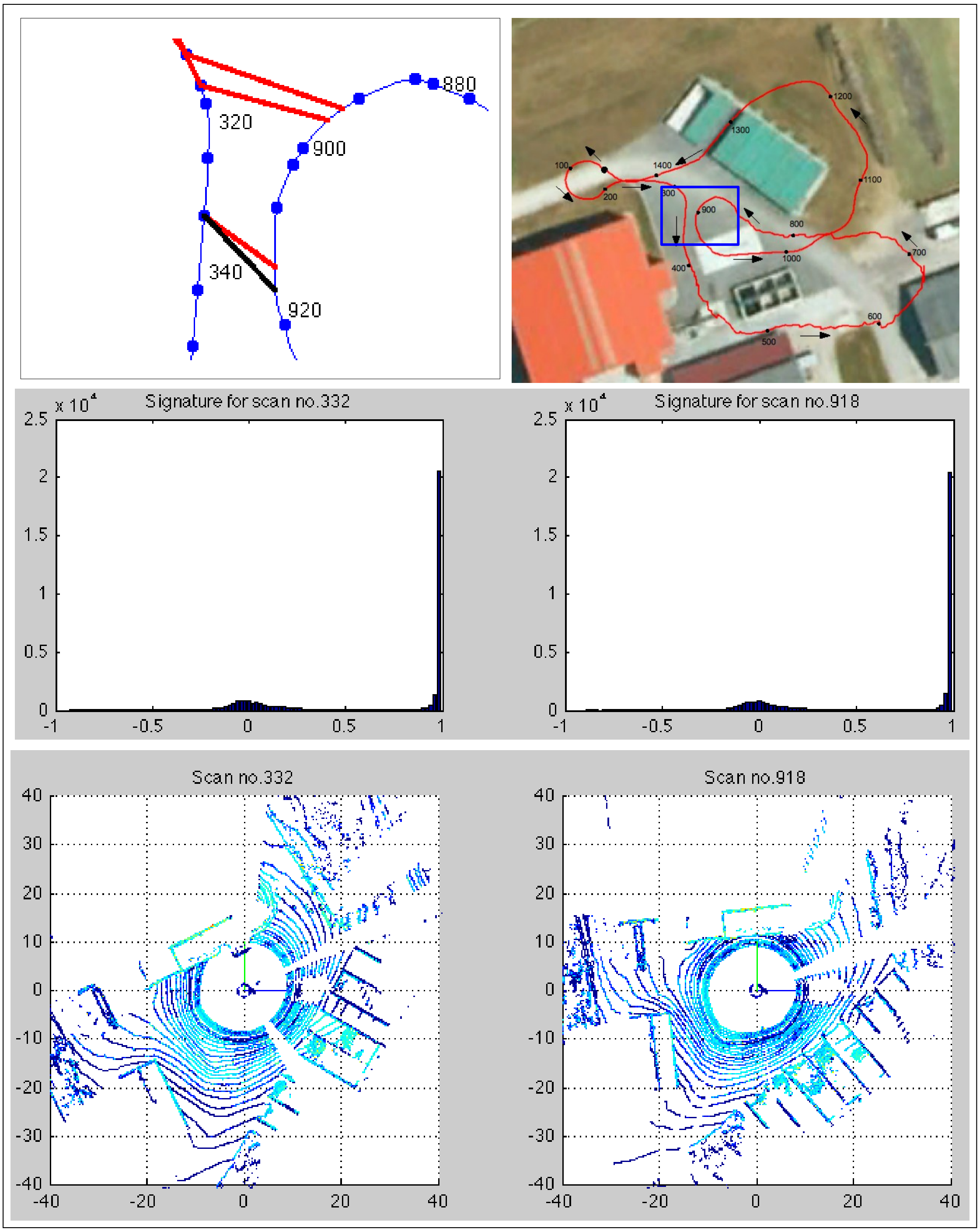

Figure 5.14: Loop closures as the robot moves through positions 893 through 919.

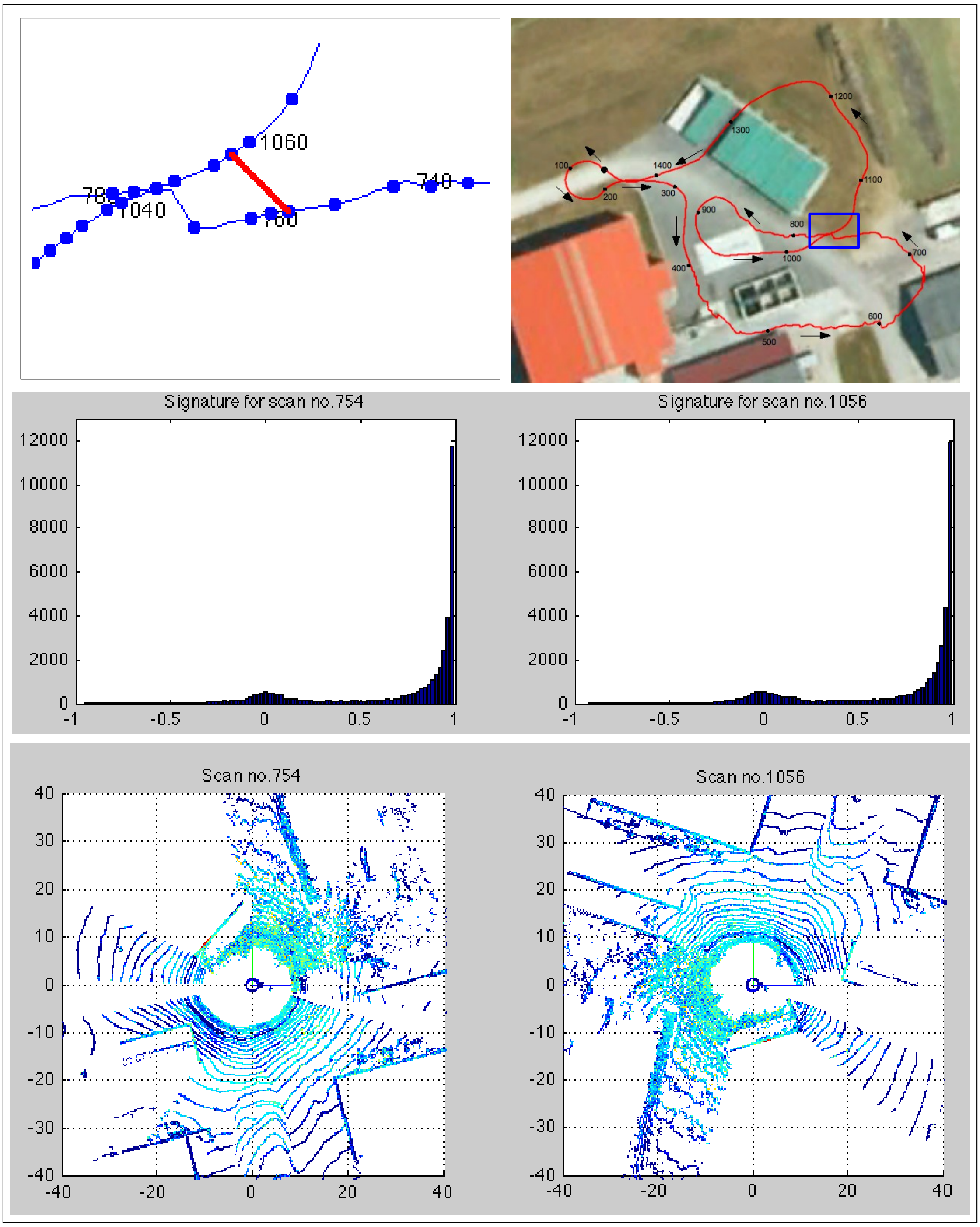

Figure 5.15: Loop closures detection between the positions 1056 and 754.

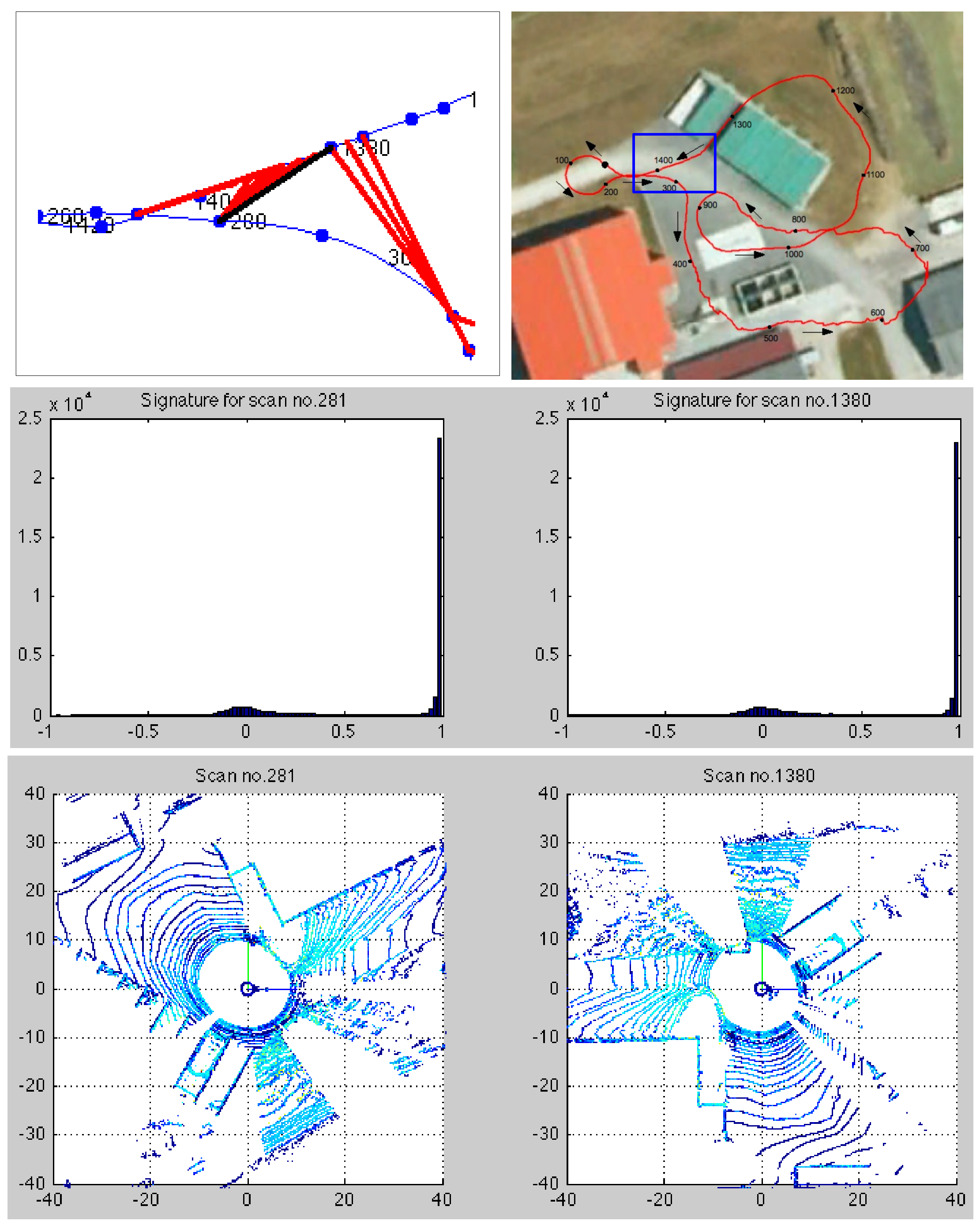

Figure 5.16: Loop closures detections as the robot moves from position 1375 through 1397.

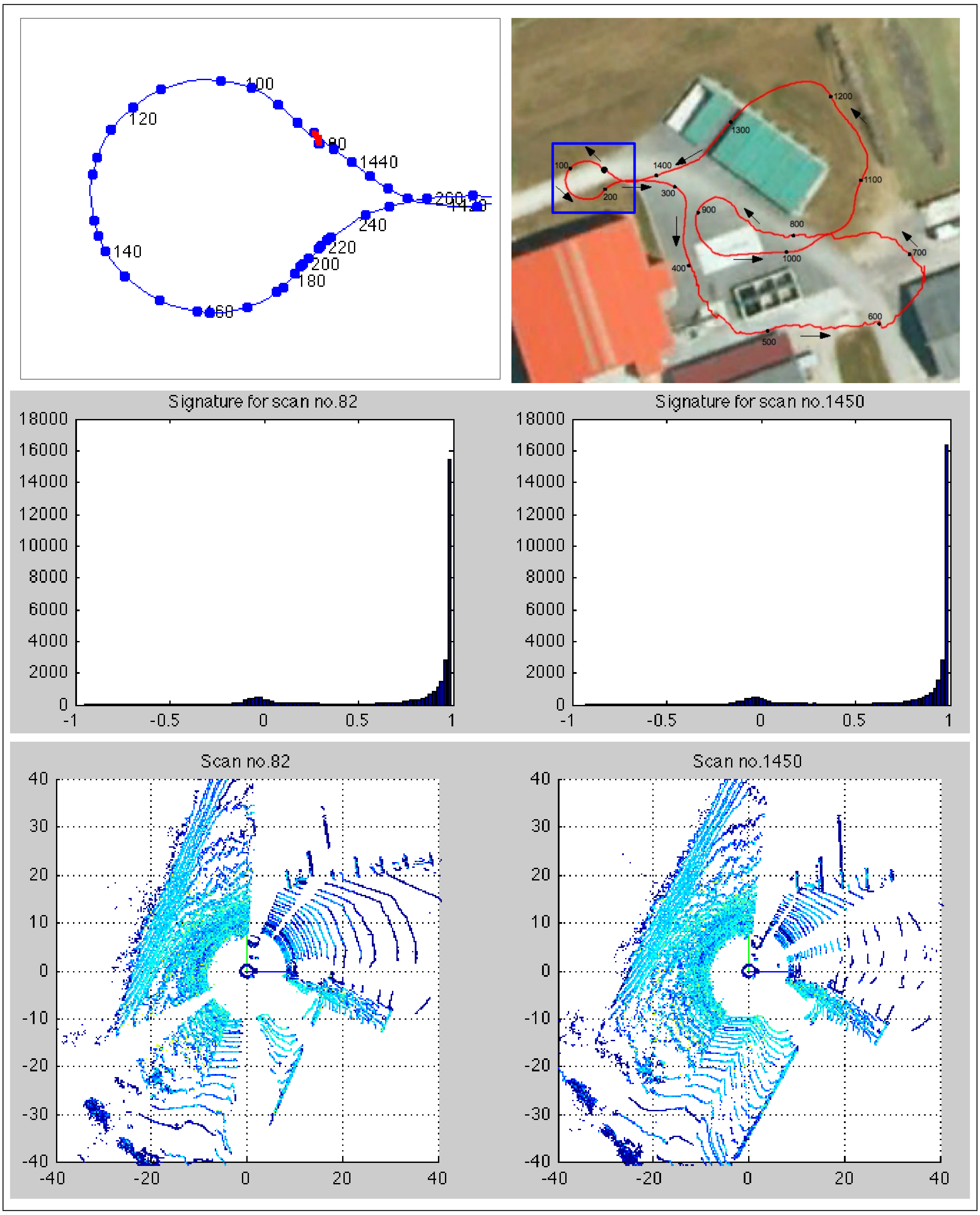

Figure 5.17: Loop closures detected at start and end positions of the robot trajectory.

544 (shown as the black line in the zoomed-in view). The local-surface-normal-based global signatures essentially encode the structure in the surroundings of the robot. One reason for these false loop closure detections is that the two positions (between which the false positive occurs) in the environment have similar structure *i.e.* similar amount of walls and ground areas etc. This similarity in the amount of structure is apparent from the two signatures shown in the figure (middle) which seem almost identical in appearance.

5.3 Determining the robot orientation

The previous section presents a method for loop closure detection or place recognition in a more general sense. This section presents a technique to efficiently determine the difference in robot orientation between the current and the loop-closure location.The technique is explained as follows.

Figure 5.19 shows the 3D point cloud acquired by one (out of 64) laser beam with 2° pitch angle at the positions 754 and 1056 of the 1505-scan data set presented in subsection 5.2.2. The figure essentially shows a single-beam subset of the 3D scans shown figure 5.15 (bottom). The robot orientation information from such single-beam scans can be encoded into small orientation-signatures as follows. The 3D point cloud acquired by the single-beam are converted from Cartesian (x, y, z) to cylindrical (θ, ρ, z) coordinates. The points beyond 50m depth are discarded. The cylindrical-coordinate data is then encoded into 360-bin signature for values of θ ranging from 0 to 360° where each bin contains the mean $\rho - coordinate$ of 3D points that lie within the values of θ corresponding to that bin. The orientation-signatures for the scans of figure 5.19 are shown in figure 5.20.

Let's denote any two orientation-signatures from two nearby locations as S_i and S_j. In order to determine the difference in robot orientation between the two locations, the sum of absolute difference between the two signatures is computed for 360 iterations as follows. S_i is kept stationary while S_j is moved by one bin in a circular fashion for 360 iterations, and a sum of element-to-element absolute differences is computed at each iteration. The iteration which has the lowest sum result represents the rotation around $z - axis$ that scan corresponding to S_j has to undergo in order to be aligned with scan corresponding to S_i. Figure 5.21 shows the curve formed by performing the above

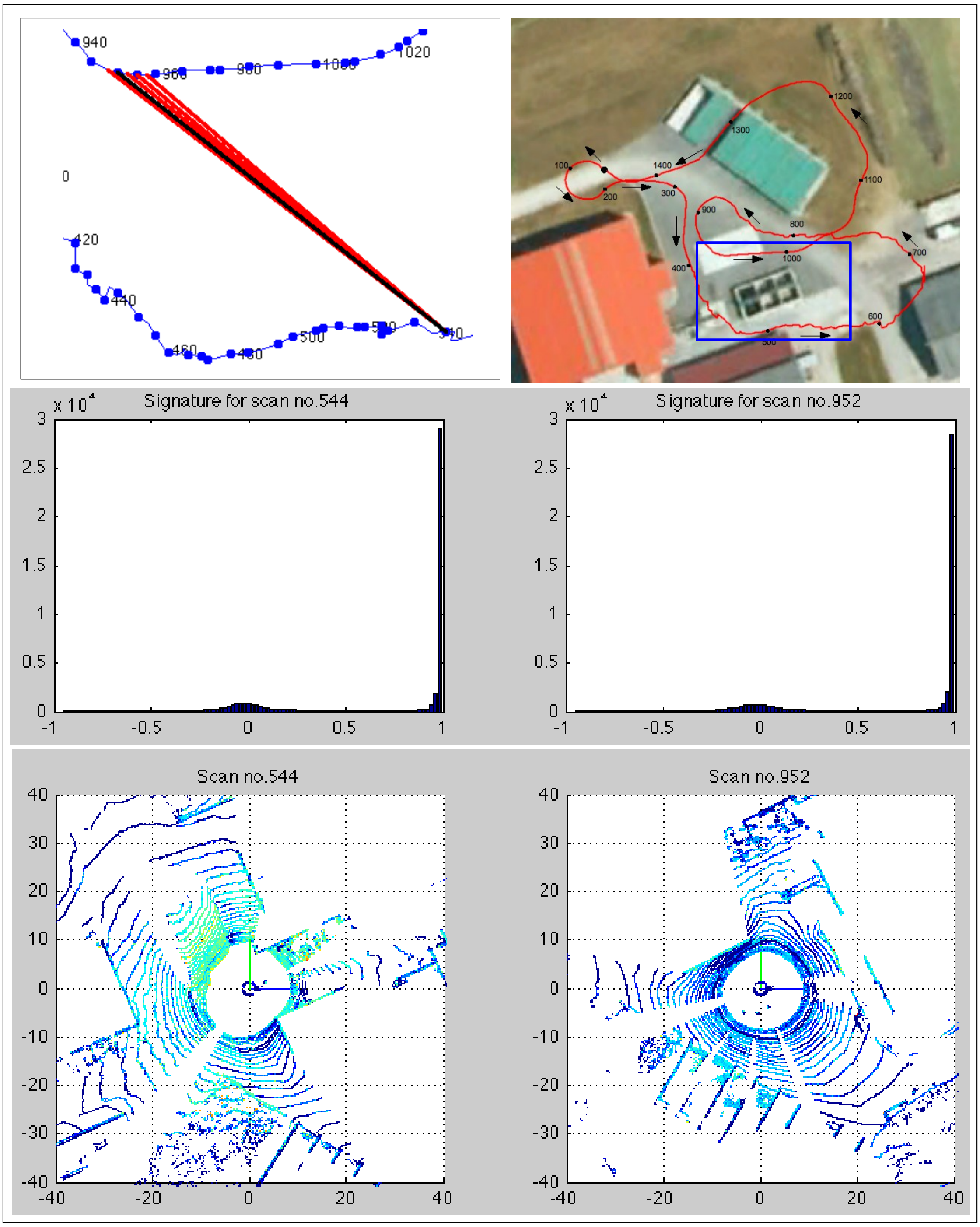

Figure 5.18: False positive loop closures detection.

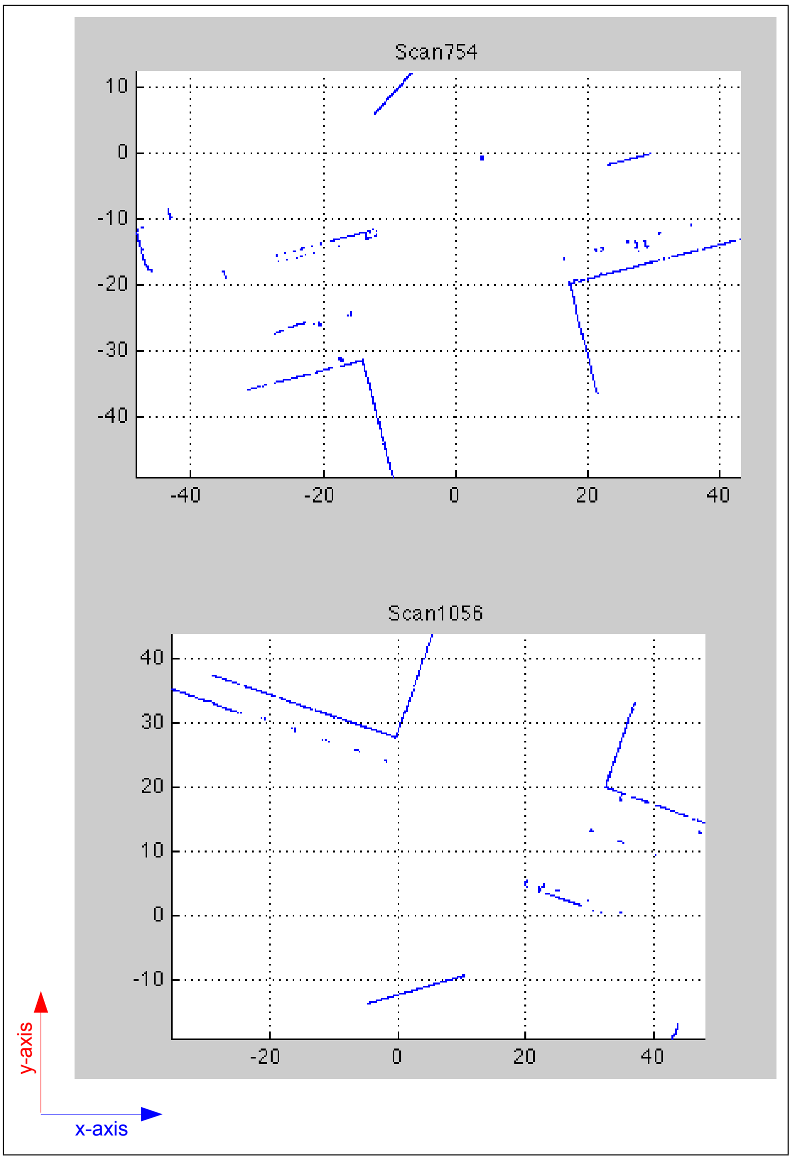

Figure 5.19: 3D point clouds for locations 754 and 1056 acquired by one (of the 64) laser beam in the Velodyne lidar device.

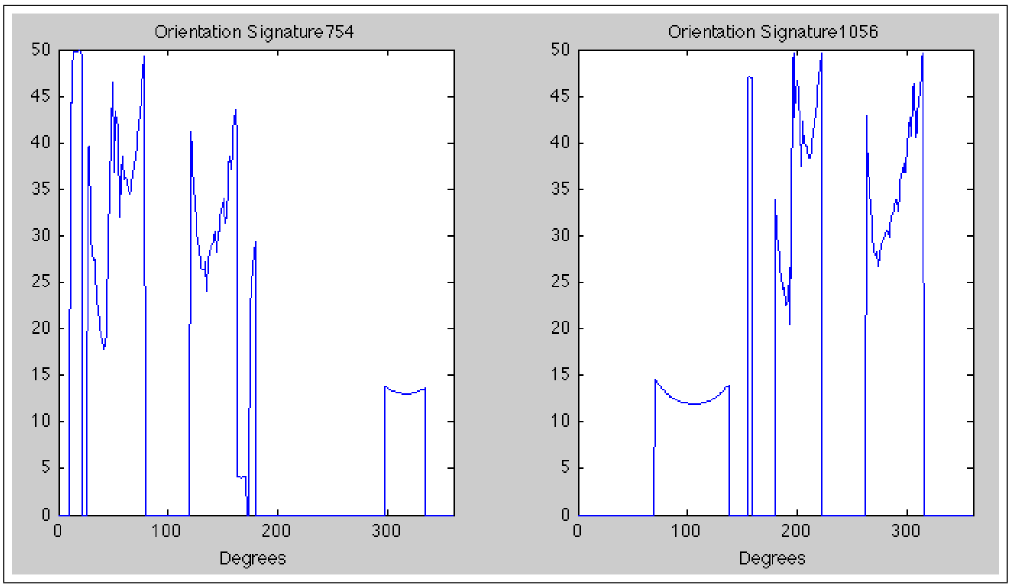

Figure 5.20: Orientation signatures for the single-beam scans shown in figure 5.19.

mentioned orientation-signature comparison for locations 754 and 1056. The curve has its minimum value at 218°. From the inspection of figure 5.19 it can be verified that scan 1056 has to undergo an approximate 220° rotation around the $z - axis$ in order to be aligned to the scan 754.

Figures 5.22, 5.23 and 5.24 show the single-beam scans, the corresponding orientation-signatures and the comparisons curves for the location pairs 332-918, 281-1380 and 82-1450 respectively. These three location pairs have already been shown in the loop-closure detection context in the figures 5.14, 5.16 and 5.17. From inspection of the figures 5.22, 5.23 and 5.24 it can be seen that the minima of the orientation-comparison curves indicate correct orientation difference between the respective scan pairs.

Once the robot orientation difference is known, a one way of estimating a precise 3D transformation between two robot positions is by applying iterative closet point algorithm or one of it's variants such as Generalized-ICP presented in (74) to the point cloud data from all or a subset of laser beams.

5.4 Discussion

The ability of a mobile robot to correctly localize itself is of paramount importance. During autonomous operations, a robot has to build an environment model by perceiv-

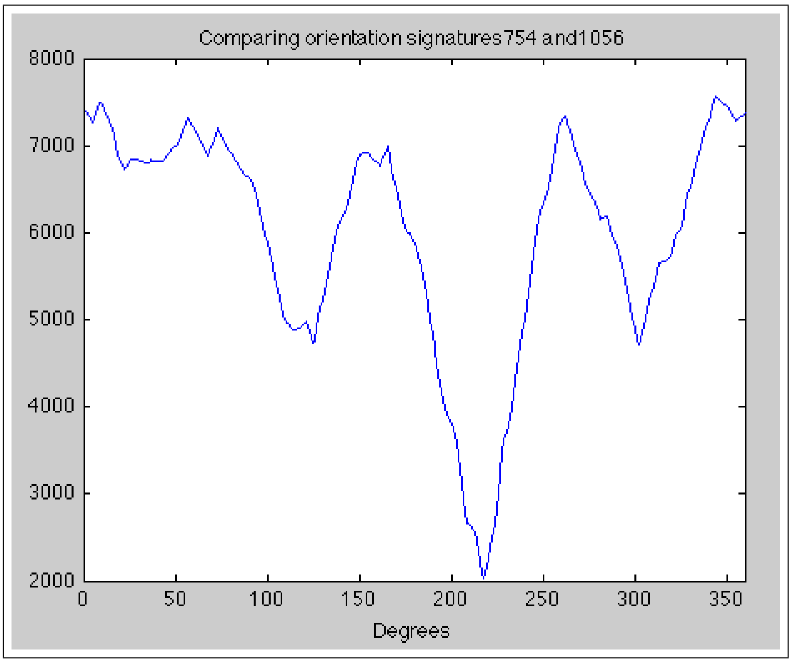

Figure 5.21: Comparison curve between the orientation-signatures for locations 754 and 1056. The curve has the minimum value at 218° implying that the scan 1056 has to undergo a rotation of 218° around z-axis in order to be aligned to scan 754 (*cf.* figure 5.19).

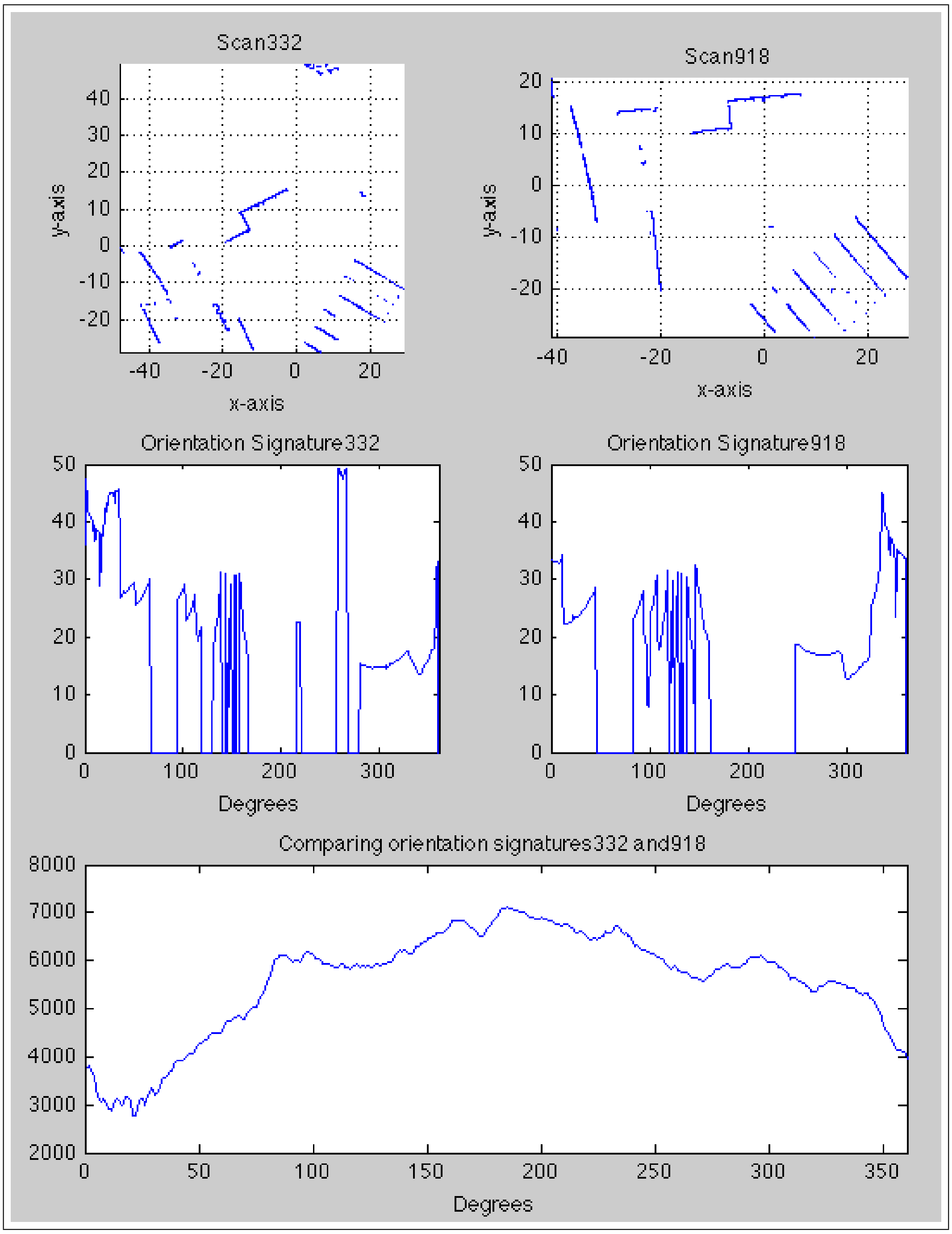

Figure 5.22: Determining the orientation difference between locations 332 and 918. The comparison curve (bottom) has its minimum at 22°.

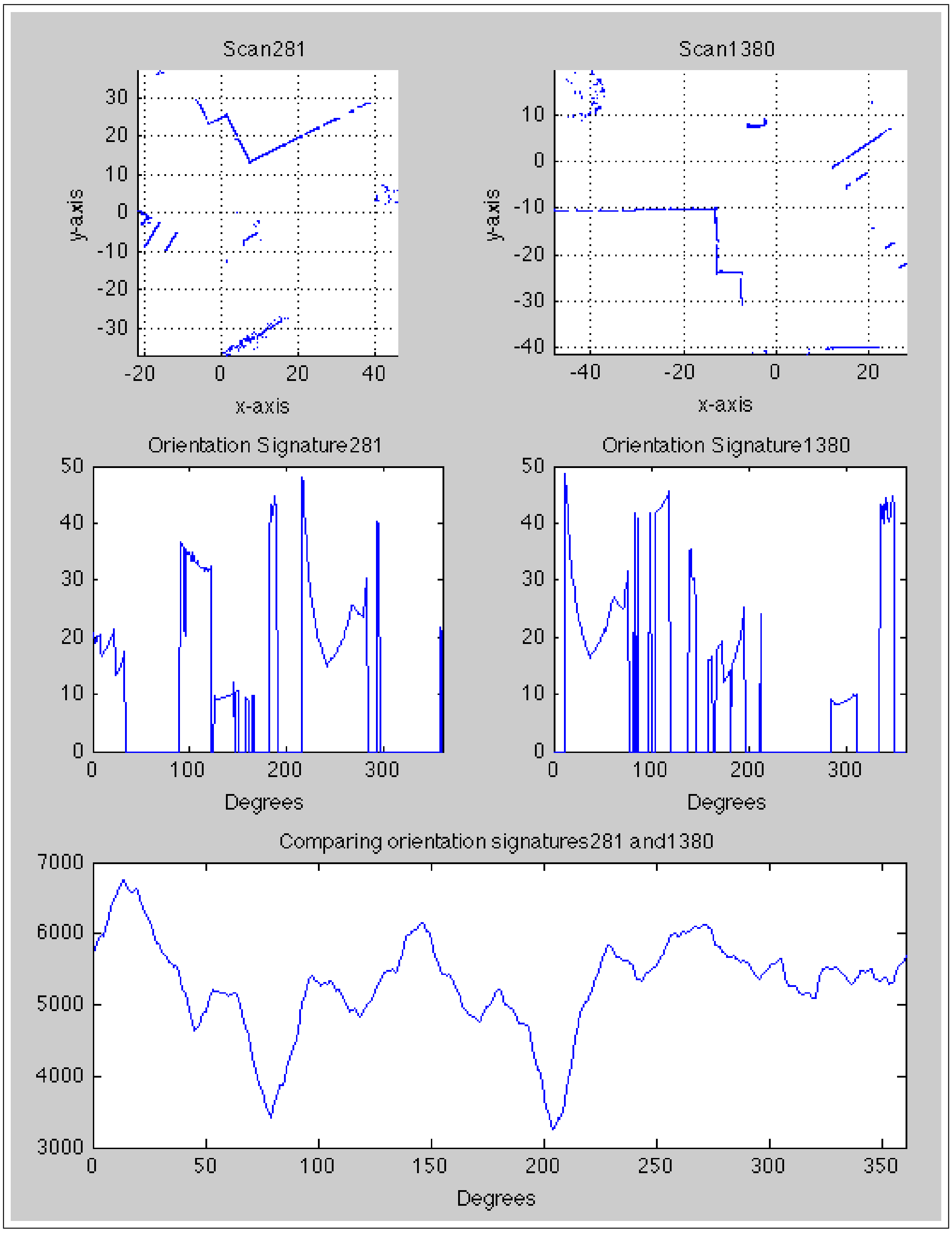

Figure 5.23: Determining the orientation difference between locations 281 and 1380. The comparison curve (bottom) has its minimum at 204°.

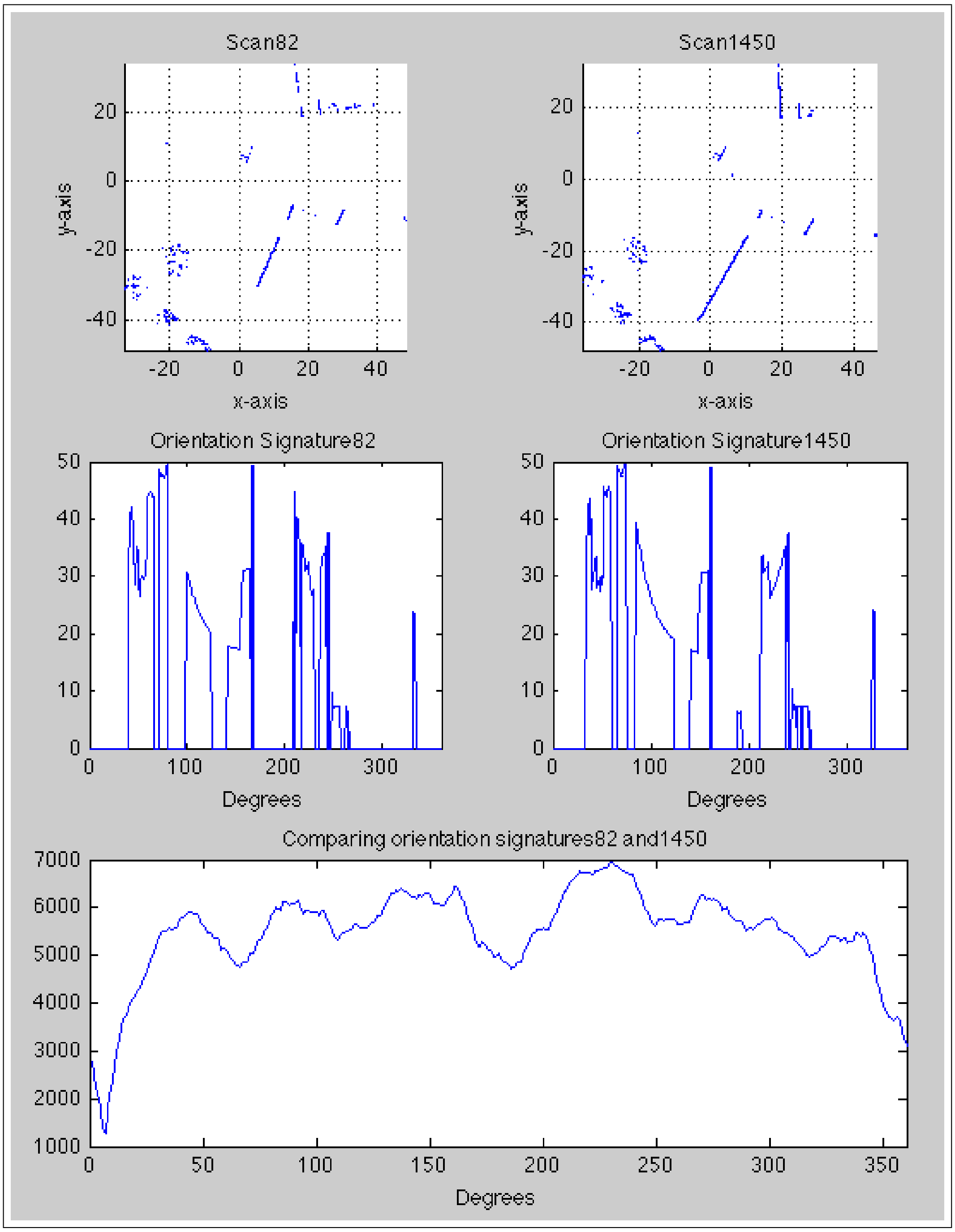

Figure 5.24: Determining the orientation difference between locations 82 and 1450. The comparison curve (bottom) has its minimum at 7°.

ing its surroundings and without reliable localization estimates the robot cannot build spatially consistent environment models. Different solutions to the localization problem have their own requirements and characteristics and hence need to be integrated for robust localization during autonomous navigation.

This chapters has presented a technique for qualitative view-based localization using small-sized signatures extracted from rich multi-beam lidar data. The proposed technique works in a two step process. The first step is essentially a "place recognition" framework. It is performed by indexing global-signatures extracted from the lidar data. As the robot moves, the newly extracted signatures are compared with the indexed signatures in order to detect the locations that have already been visited by the robot. Once the robot recognizes that it is in an already visited area, the relative orientation of the current and a previously visited robot position is determined by comparing compact orientation-signatures for the two locations. This constitutes the second step on the proposed technique.

On the one hand the proposed technique can act as an independent view-based localization system during autonomous navigation. After place recognition and determination of relative orientation, iterative closest point algorithm can be employed in order to estimate a precise 3D transformation between the current and the already visited closest location. On the other hand the proposed technique can work in combination with a SLAM system running onboard a robot. While performing feature-based SLAM, and especially while performing loop-closures, one of the most challenging problems is the data association. In order to perform loop-closures, a robot has to identify that it is in fact in an already visited area and the features that it is observing belong to the landmarks already present in its map at an already visited location. The view-based localization technique presented in this chapter can assist the SLAM system by identifying the potential loop-closure locations and thus easing the data association problem for the SLAM system.

Integrating various solutions to the localization problem to work collaboratively is the key to reliable localization during autonomous navigation. The view-based localization technique presented in this chapter in combination with other localization solutions takes robust autonomous navigation one step forward.

Chapter 6

Conclusions

The introduction of novel multi-beam lidar devices that quickly acquire dense 3D point clouds has revolutionized the way robots perceive their surroundings. The introduction of such devices calls for the development of techniques that exploit these sensors to their full potential in robotics. This thesis addresses the topic of exploitation of such devices in autonomous outdoor navigation. Specifically, the intrinsic calibration of such devices and a technique for qualitative view-based localization by extracting small-sized global signatures from rich lidar data have been presented in the thesis.

6.1 Summary

The first part of this thesis consists of chapters 2 and 3, and addresses the topic of multi-beam lidar calibration. In **chapter 2** an overview of lidar sensors used in robotics has been presented. The chapter also introduces the multi-beam lidar device Velodyne HDL-64E S2 and presents its use in outdoor autonomous robotics to date. We argue that lidar sensors in general, and especially the novel-multi beam lidar sensors, have a great potential for applications in robotics and especially in outdoor navigation because of several advantages they offer over other exteroceptive sensors. These advantages include fast and direct acquisition of 3D data with fairly long operational ranges (for instance 50m for low reflectivity objects for the Velodyne HDL-64E S2) and superior range measurement accuracy in comparison to stereovision for instance. We believe that multi-beam lidar sensors have a great potential for autonomous outdoor navigation in the the years to come.

In order to exploit the perception ability of multi-beam lidar sensors, the most basic requirement is their accurate calibration. In **chapter 3** a technique for calibrating the geometric and distance correction parameters for multi-beam lidar sensors has been presented. The technique is based on the optimization of calibration parameters by comparing acquired point cloud data to the ground-truth geometrical shape of a calibration target (a planar wall). Results have been presented for the calibration of orientation (rotational and vertical correction angles) and distance correction factors for 64 laser beams of the Velodyne HDL-64E S2 lidar device. The results show that in comparison to factory provided calibration parameters, recalibration improves the acquired point cloud quality in terms of range measurements confirming to the ground truth distance of objects from the lidar. Recalibration also improves the point cloud sharpness (measured as the thickness of planar structures in the scene) in most of the areas in the scene, but this is not the case for each and every object present in the scene. Although the presented calibration technique brings improvement in the device calibration in comparison to the factory calibration, there is still room for improvements of the calibration of such devices.

The second part of this thesis consists of chapter 4 and 5 and addresses the topic of qualitative localization using multi-beam lidar devices. In **chapter 4** a discussion on the importance of reliable localization during autonomous operation of mobile robots has been presented. An overview of the various solutions employed to solve the localization problem in autonomous robotics has also been presented including an overview of view-based localization techniques using local and global signatures. A technique for qualitative view-based localization by extracting small-sized global signatures from multi-beam lidar data, such as from the Velodyne, has been presented in **chapter 5**. Such sensors provide huge amounts of 3D data (1.33 million point per second for the Velodyne) and therefore processing and storing all the 3D data in its raw form requires a lot of memory and computation. This makes the extraction and use of small-sized global signatures especially interesting. The said signatures are based on the histograms of local surface normal information which can efficiently be extracted from the 3D data by exploiting the arrangement of laser-beams inside the multi-beam lidar device. The presented view-based localization technique is based on indexing and comparison of the said global signatures. The technique consists of a first “place recognition” or “loop-closure detection” step. This step is then followed by a second step that estimates the relative orientation between a current and a previously visited location. Experimental results on Velodyne data have been presented that validate the proposed techniques.

6.2 Contributions

The principal area of contribution of this thesis is the exploitation of novel multi-beam 3D lidar sensors for autonomous outdoor navigation. More specifically the thesis contributions are:

- An analysis of the characteristics and behavior of the multi-beam lidar device Velodyne HDL-64E S2.
- A technique for the calibration of geometric and distance correction parameters of a multi-beam lidar device.
- A comparison of two methods for extracting local surface normal information from multi-beam lidar data *i.e.* (i) by performing plane fitting in local neighborhoods and (ii) by exploiting the arrangement of laser beams inside the lidar device.
- A technique for the extraction of small-sized (679:1 data reduction ratio) global signatures from 3D point clouds, along with a method to quantitatively compare them. The signatures are based on the histograms of local surface normal information extracted from the point cloud.
- A technique for qualitative view-based localization by indexing and comparing the above mentioned global signatures, complemented with an approach to determine the relative orientation between two close robot positions.

6.3 Future work

Reliable autonomous outdoor navigation remains a very challenging task. It has been a very active research area for decades, and it still is. The introduction of novel multi-beam lidar devices with their fast and rich data acquisition has enabled outdoor robots to perceive their surroundings like never before. The exploitation of these sensors to their full potential remains an open area of research. Some possible extensions to the work presented in this thesis on the exploitation of these sensors for autonomous navigation and listed below:

- **Richer global signatures:** The global signatures presented in chapter 5 are based only on the local-surface-normal information. The signature definition can be enriched by incorporating more information from the scene e.g. depth or height of 3D points etc. Richer signature definition can increase the robustness of the view-based localization performed using such signatures. The global signatures defined in the thesis are rotation invariant. Directional information from 360° Velodyne scans can be incorporated into the signatures in order to readily estimate relative orientation between two given scans while performing view-based localization.

- **Map-based localization:** The global signatures presented in this thesis can be used localize a robot within a map that is a priori available to the robot. Using the map, one can pre-compute thanks to simulations the global signature for numerous locations in the map. In this way the robot, at a given time, can compare the currently extracted actual signature from the environment to the simulated ones in order to localize itself within the map.

- **Local feature extraction:** The extraction of local features such as points, lines, and planes from Velodyne data should be investigated. These local features can be used for performing view-based localization or SLAM for instance. Planar features can be especially interesting in an air-ground multi-robot context. The planar features can be used to identify common landmarks seen by a ground robot and by a UAV (unmanned aerial vehicle) while performing multi-robot SLAM for instance.

- **Lidar calibration:** As mentioned in chapter 3, there is still room for improvement in the techniques for multi-beam lidar calibration. The planar-calibration-target-based calibration technique presented in chapter 3 can be extended to automatic identification and segmentation of possible calibration targets that are present in a point cloud. The technique can further be investigated for other types of objects that can be used as calibration targets – poles and depth gradients along straight lines for instance.

References

[1] SICK AG. **LMS5xx laser measurement sensors** [online]. Available from: `http://www.sick.com/group/en/home/products/product_news/laser_measurement_systems/pages/lms5xx_laser_measurement_sensors.aspx` [cited 23 September 2011]. 38

[2] Henrik Andreasson and Tom Duckett. **Topological Localization for Mobile Robots using Omni-directional Vision and Local Features**. In *5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, July 2004. 86

[3] Adrien Angeli, David Filliat, Stephane 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, October 2008. 75, 87

[4] Applanix. **POS LV** [online]. Available from: `http://www.applanix.com/products/land/pos-lv.html` [cited 4 November 2011]. 73

[5] Freie Universität Berlin Artificial Intelligence Group. **Autonomos Labs** [online]. Available from: `http://autonomos.inf.fu-berlin.de/` [cited 22 August 2011]. 34

[6] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, and P. Vernaza. **Little Ben, The Ben Franklin Racing Team's Entry in the 2007 DARPA Urban Challenge**. *Journal of Field Robotics*, **25**:598–614, 2008. 21, 24, 34

[7] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab*. Available from: `http://www.vision.caltech.edu/bouguetj` [cited 22 Feb 2010]. 19

[8] M. Carreras, P. Ridao, R. Garcia, and T. Nicosevici. **Vision-based Localization of an Underwater Robot in a Structured Environment**. In *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003. 78, 79

[9] Sung-Hyuk Cha. **Taxonomy of nominal type histogram distance measures**. In *American Conference on Applied Mathematics*, pages 325–330. World Scientific and Engineering Academy and Society (WSEAS), 2008. 102

[10] Baptiste Charmette, Eric Royer, and Frédéric Chausse. **Efficient planar features matching for robot localization using GPU**. In *Embedded Computer Vision Workshop*, San Francisco, CA, USA, June 2010. 86

[11] Hakyoung Chung, Lauro Ojeda, and Johann Borenstein. **Accurate Mobile Robot Dead-reckoning With a Precision-calibrated Fiber Optic Gyroscope**. *IEEE Transactions on Robotics and Automation*, **17**:80–84, 2001. 72

[12] Google Code. **The making-of ”House of Cards”** [online]. Available from: `http://code.google.com/creative/radiohead/` [cited 22 August 2011]. 37

[13] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. **Visual categorization with bags of keypoints**. In *Workshop on Statistical Learning in Computer Vision (ECCV)*, pages 59–74, Prague, Czech Republic, 2004. 86

[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. 75, 87

[15] Christoph Dold and Claus Brenner. **Registration of terrestrial laser scanning data using planar patches and image data**. In *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, September 2006. 91

[16] Michael Dunbar and Michael Neuman. **Mobile mapping and data collection**. *GEOconnexion Internation Magazine*, April 2008. 37

[17] Ethan Eade and Top Drummond. **Edge Landmarks in Monocular SLAM**. In *British Machine Vision Conference*, Edinburgh, UK, 2006. 74

[18] Pantelis Elinas, Robert Sim, and James J. Little. **-SLAM: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution**. In *IEEE International Conference on Robotics and Automation*, Orlando, FL, USA, 2006. 74

[19] Chris Urmson et al. **Tartan Racing: A multi-modal approach to the DARPA Urban Challenge**. Technical report, Carnegie Mellon University, General Motors, Caterpillar and Continental AG, April 2007. x, 31, 32, 33

[20] Michael Montemerlo et al. **Junior: The Stanford Entry in the Urban Challenge**. *Journal of Field Robotics*, **25**(569-597), September 2008. 32, 35

[21] David Filliat and Jean-Arcady Meyer. **Map-based navigation in mobile robots. – I. A review of localization strategies**. *Cognitive Systems Research*, **4**(4):243–282, December 2003. 80

[22] Craig Glennie and Derek D. Lichti. **Static calibration and analysis of the Velodyne HDL-64E S2 for high accuracy mobile scanning**. *Remote Sensing*, **2**:1610–1624, June 2010. x, 21, 58, 65, 66

[23] Ibeo Automotive Systems GmbH. *ibeo LUX*. Merkurring 60-62, 22143 Hamburg, Germany. 16

[24] José-Joel Gonzalez-Barbosa and Simon Lacroix. **Rover localization in natural environments by indexing panoramic images**. In *IEEE International Conference on Robotics and Automation*, Washington DC, USA, May 2002. 85, 102

[25] Google. **What we're driving at** [online]. Available from: `http://googleblog.blogspot.com/2010/10/what-were-driving-at.html` [cited 22 August 2011]. 34

[26] Anders Hagnelius. *Visual Odometry.* Master's thesis, Department of Computing Science, Umeå University, SE-901 87 Umeå, Swedenß, April 2005. 73

[27] David S. Hall and Bruce S. Hall. **Team DAD technical paper**. Technical report, Velodyne Acoustics Inc., August 2005. 16, 31

[28] Ryan Halterman and Michael Bruch. **Velodyne HDL-64E LIDAR for Unmanned Surface Vehicle Obstacle Detection**. In *SPIE Unmanned Systems technology XII*, Orlando, Florida, USA, April 2010. x, 37

[29] Hokuyo. **Scanning range finder UTM-30LX** [online]. Available from: `http://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html` [cited 11 August 2011]. 15

[30] Andrew Howard, Sajid Siddiqi, and Gaurav S. Sukhatme. **An Experimental Study of Localization Using Wireless Ethernet**. In *The 4th International Conference on Field and Service Robotics*, July 2003. 77

[31] Patric Jensfelt and Steen Kristensen. **Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking**. *IEEE Transactions on Robotics and Automation*, **17**(5):748–760, October 2001. 80

[32] Matjaž Jogan, Aleš Leonardis, Horst Wildenauer, and Horst Bischof. **Mobile Robot Localization Under Varying Illumination**. In *International Conference on Pattern Recognition*, Los Alamitos, CA, USA, 2002. 85

[33] Michael Kaess and Frank Dellaert. **Visual SLAM with multi-camera rig**. Technical Report GIT-GVU-06-06, Georgia Institute of Technology, February 2006. ix, 4

[34] T. Kahlmann and H. Ingensand. **Calibration for increased accuracy of the range imaging camera Swissranger**. In *IAPRS*, **XXXVI**, pages 136–141, September 2006. 21

[35] Soeren Kammel, Benjamin Pitzer, Stefan Vacek, Joachim Schroeder, Christian Frese, Moritz Werling, and Matthias Goebl. **DARPA Urban Challenge Team AnnieWAY Technical System Description**. Technical report, Cognitive Automobiles, 2007. 34

[36] Jae-Hean Kim and Myung Jin Chung. **SLAM with omni-directional stereo vision sensor**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, USA, 2003. 74

[37] Jungho Kim, Kuk-Jin Yoon, Jun-Sik Kim, and Inso Kweon. **Visual SLAM by single-camera catadioptric stereo**. In *International Joint Confetence SICE-ICASE*, Busan, Korea, 2006. ix, 3, 5

[38] Konica-Minolta. **Konica Minolta RANGE5** [online]. Available from: `http://www.konicaminolta.com/sensingusa/products/3D-Scanning/3D-Scanners/range5/specifications.html` [cited 12 August 2011]. 18

[39] Thomas Lemaire. *Simultaneous Localisation And Mapping with Monocular Vision.* PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, December 2006. 74

[40] Thomas Lemaire, Cyrille Berger, Il-Kyun Jung, and Simon Lacroix. **Vision-Based SLAM: Stereo and Monocular Approaches**. *International Journal of Computer Vision*, **74**:343–364, 2007. 74

[41] Thomas Lemaire and Simon Lacroix. **SLAM with panoramic vision**. *Journal of Field Robotics*, **24**:91–111, 2007. 74

[42] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. **Map-Based Precision Vehicle Localization in Urban Environments**. In *Robotics: Science and Systems*, Atlanta, GA, USA, June 2007. 79

[43] Jesse Levinson and Sebastian Thrun. **Unsupervised Calibration for Multi-beam Lasers**. In *12th International Symposium on Experimental Robotics*, December 2010. 22, 92

[44] Yangming Li and Edwin B. Olson. **Extracting general-purpose features from LIDAR data**. In *IEEE International Conference on Robotics and Automation*, May 2010. 90

[45] Yangming Li and Edwin B. Olson. **A General Purpose Feature Extractor for Light Detection and Ranging Data**. *Sensors*, 2010. 90

[46] M. Lindner and A. Kolb. *Lateral and Depth Calibration of PMD-Distance Sensors*, **4292/2006** of *Lecture Notes in Computer Science*, pages 524–533. Springer Berlin/Heidelberg, 2006. 21

[47] David G. Lowe. **Object Recognition from Local Scale-Invariant Features**. In *International Conference on Computer Vision*, Corfu, Greece, 1999. 83

[48] Richard Mason, Jim Radford, Robert Walters, David Caldwell, Bill Caldwell, and Dmitriy Kogan. **DARPA Urban Challenge The Golem Group LLC**. Technical report, The Golem Group LLC, April 2007. 34

[49] The Mathworks Inc. *Matlab Optimization Function: fmincon*. 53

[50] Toshihiro Matsui, Hideki Asoh, and Simon Thompson. **Mobile Robot Localization Using Circular Correlations of Panoramic Images**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000. 84

[51] Yoshio Matsumoto, Masayuki Inaba, and Hirochika Inoue. **Visual Navigation using View-Sequenced Route Representation**. In *IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, USA, April 1996. 84

[52] Yoshio Matsumoto, Masayuki Inaba, and Hirochika Inoue. **View-based navigation using an omniview sequence in a corridor environment**. *Machine Vision Applications*, **14**:121–128, 2003. xi, 75, 84, 85

[53] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. **Image-Based Monte-Carlo Localisation with Omnidirectional Images**. *Robotics and Autonomous Systems*, **48**:17–30, 2004. 80

[54] MESA Imaging AG, Technoparkstrasse 1, 8005 Zurich, Switzerland. *SR4000 Data Sheet*, May 2011. 17

[55] Michael Milford and Gordon Wyeth. **Featureless vehicle-based visual SLAM with a consumer camera**. In *Australasian Conference on Robotics and Automation*, Brisbane, Queensland, Australia, 2007. xi, 83

[56] Michael Montemerlo, Sabestian Thrun, Daphne Koller, and Ben Wegbreit. **FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem**. In *AAAI National Conference on Artificial Intelligence*, pages 593–598, 2002. 74

[57] Frank Moosmann, Oliver Pink, and Christoph Stiller. **Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion**. In *IEEE Intelligent Vehicles Symposium*, June 2009. 92

[58] Andreu Corominas Murtra. **Thesis Project: Active Map-based Localization for Cooperative Mobile Robots and Urban Environments**. Technical report, Universitat Politècnica de catalunya, September 2007. 80

[59] Luis E. Navarro-Serment, Christiaan J.J. Paredis, and Pradeep K. Khosla. **A Beacon System for the Localization of Distributed Robotic Teams**. In *International Conference on Field and Service Robotics*, pages 232–237, 1999. 4, 6, 77

[60] Viet Nguyen, Stefan Gächter, Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. **A comparison of line extraction algorithms using 2D range data for indoor mobile robotics**. *Autonomous Robots*, **23**:97–111, 2007. 90

[61] David Nistér, Oleg Naroditsky, and James Bergen. **Visual odometry for ground vehicle applications**. *Journal of Field Robotics*, **23**:3–20, 2006. 72, 73

[62] Lauro Ojeda and Johann Borenstein. **Improved Position Estimation for Mobile Robots on Rough Terrain Using Attitude Information**. Technical report, Department of Mechanical Engineering, The University of Michigan, USA, August 2001. 72

[63] Velijo Otsason. *Accurate Indoor Localization Using Wide GSM Fingerprinting.* Master's thesis, Institute of Computer Science, University of Tartu, Estonia, 2005. 77

[64] Anna Petrovskaya and Sebastian Thrun. **Efficient Techniques for Dynamic Vehicle Detection**. In *International Symposium on Experimental Robotics*, pages 79–91, 2008. 34

[65] Anna Petrovskaya and Sebastian Thrun. **Model Based Vehicle Tracking for Autonomous Driving in Urban Environments**. In *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008. 34

[66] Arnau Ramisa, Adriana Tapus, David Aldavert, Ricardo Toledo, and Ramon López de Mántaras. **Robust Vision-Based Localization using Combinations of Local Feature Regions Detectors**. *Autonomous Robots Journal*, **27**:373–385, 2009. 86

[67] REIGL Laser Measurement Systems GmbH, A-3580 Horn, Austria. *Laser distance, level and speed sensor LD90-3*, 2006. ix, 14

[68] Riegl. **Reigl LPM-321** [online]. Available from: `http://riegl.com/nc/products/terrestrial-scanning/produktdetail/product/scanner/2/` [cited 12 August 2011]. 18

[69] Gerald P. Roston and Eric P. Krotkov. **Dead Reckoning Navigation for Walking Robots**. Technical Report CMU-RI-TR-91-27, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA, November 1991. 73

[70] R. Rouveure, M. O. Monod, and P. Faure. **Mapping of the environment with a high resolution ground-based radar imager**. In *IEEE Mediterranean Electrotechnical Conference*, pages 822–828, Ajaccio, France, May 2008. 5

[71] Thales Optronique SA. **ANR 2RT3D, Rapport d'essais de qualification du Velodyne HDL64-E**. Technical report, Thales, 2010. x, 23, 29, 30, 44

[72] Shigeru Saito, Atsushi Hiyama, Tomohiro Tanikawa, and Michitaka Hirose. **Indoor Marker-based Localization Using Coded Seamless Pattern for Interior Decoration**. In *IEEE Virtual Reality Conference*, Charlotte, North Carolina, USA, March 2007. 78

[73] Joaquim Salvi, Xavier Armangué, and Joan Batlle. **A comparative review of camera calibrating methods with accuracy evaluation**. *Pattern Recognition*, **35**:1617–1635, 2002. 19

[74] Aleksandr V. Segal, Dirk Haehnel, and Sabestian Thrun. **Generalized-ICP**. In *Robotics: Science and Systems*, 2009. 119

[75] Daisuke Sekimori and Fumio Miyazaki. **Precise Dead-Reckoning for Mobile Robots Using Multiple Optical Mouse Sensors**. *Informatics in Control Automation and Robotics II*, pages 125–151, 2007. 73

[76] Mark Sheehan, Alastair Harrison, and Paul Newman. **Automatic Self-Calibration Of A Full Field-Of-View 3D n-Laser Scanner Field-Of-View 3D n-Laser Scanner**. In *12th International Symposium on Experimental Robotics*, December 2010. 38

[77] SICK AG, Nimburger Strasse 11, 79296 Ruete, Germany. *Technical Description, LMS200/211/221/291 Laser Measurement System*, 2006. 15

[78] P. Stone, P. Beeson, T. Mericli, and R. Madigan. **DARPA Urban Challenge Technical Report**. Technical report, Austin Robot Technology, 2007. 20, 34, 44

[79] P. Sturm and S. Ramalingam. **A Generic Concept for Camera Calibration**. In *European Conference of Computer Vision*, **2**, pages 1–13, 2004. 19

[80] Michael J. Swain and Dana H. Ballard. **Color Indexing**. *International Journal of Computer Vision*, **7**:11–32, 1991. 82

[81] Stanford Racing Team. **Stanford's Robotic Vehicle "Junior:" Interim Report**. Technical report, Stanford University, Stanford, CA 94305, USA, April 2007. x, 23, 24, 34, 36

[82] Sabestian Thrun. **Robotic Mapping: A Survey**. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium.* Morgan Kaufmann, 2002. 70, 71

[83] SEBASTIAN THRUN, DIETER FOX, WOLFRAM BURGARD, AND FRANK DELLAERT. **Robust Monte Carlo Localization for Mobile Robots**. *Artificial Intelligence Journal*, 2001. 80

[84] IWAN ULRICH AND ILLAH NOURBAKHSH. **Appearance-Based Place Recognition for Topological Localization**. In *IEEE International Conference on Robotics and Automation*, pages 1023–1029, San Francisco, CA, USA, 2000. 86, 102

[85] JAMES UNDERWOOD, ANDREW HILL, AND STEVE SCHEDING. **Calibration of Range Sensor Pose on Mobile Platforms**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, SanDiego (CA), USA*, Oct. 2007. ix, 19, 20, 21

[86] JAMES P. UNDERWOOD, ANDREW HILL, THIERRY PEYNOT, AND STEVEN J. SCHEDING. **Error Modeling and Calibration of Exteroceptive Sensors for Accurate Mapping Applications**. *Journal of Field Robotics*, **27**(1):2–20, January/February 2010. 20

[87] JAMES PATRICK UNDERWOOD. *Reliable and Safe Autonomy for Ground Vehicles in Unstructured Environments*. PhD thesis, The University of Sydney, Australia, December 2008. 5

[88] VELODYNE. **Velodyne High Definition Lidar: Origins** [online]. Available from: `http://www.velodyne.com/lidar/hdlabout/origins.aspx` [cited 12 August 2011]. 16

[89] VELODYNE. **Velodyne's HDL-64E: A high definition lidar sensor for 3D applications**. Technical report, Velodyne Acoustics Inc., 345 Digital Drive, Morgan Hill, CA 95037, USA, October 2007. ix, 14

[90] VELODYNE. *HDL-64E S2 User's Manual*. Velodyne Lidar, Inc., 345 Digital Drive, Morgan Hill, CA 95037, USA, 2008. 1, 16, 24, 43, 89

[91] VELODYNE. *High Definition Lidar HDL32E User's Manual*. Velodyne Lidar Inc, 345 Digital Drive, Morgan Hill, CA 95037, USA, 2011. 16

[92] Fredrik Viksten, Klas Nordberg, and Mikael Kalms. **Point-of-Interest Detection for Range Data**. In *IEEE International Conference on Pattern Recognition*, Dec 2008. 90

[93] Niall Winters, José Gaspar, Gerard Lacey, and José Santos-Victor. **Omni-directional Vision for Robot Navigation**. In *IEEE Workshop on Omnidirectional Vision*, Washington DC, USA, 2000. 85

[94] ULand Wong. **Terrain obstacle detection and analysis using LIDAR**. 6

[95] Michael Ying Yang and Wolfgang Förstner. **Plane Detection in Point Cloud Data**. Technical report, University of Bonn, Department of Photogrammetry, Institute of Geodesy and Geoinformation, January 2010. 91

[96] Wei Zhang and Jana Kosecka. **Image Based Localization in Urban Environments**. In *International Symposium on 3D Data Processing, Visualization and Transmission*, Chapel Hill, North Carolina, USA, 2006. xi, 83, 86