Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Prof. Dr.-Ing./Univ. Tokio Martin Buss
Prof. Dr.-Ing. habil. Dirk Wollherr

3D Robotic Mapping and Place Recognition

Muhammad Sheraz Khan

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Gordon Cheng, Ph.D.
Prüfer der Dissertation:	
	1. Prof. Dr.-Ing. habil. Dirk Wollherr
	2. Prof. Dr.-Ing. Darius Burschka

Die Dissertation wurde am 20.03.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 16.11.2017 angenommen.

Foreword

This thesis presents the research work carried out by me within a period of $4\frac{1}{2}$ years under the supervision of Prof. Wollherr and Prof. Buss at the Chair of Automatic Control Engineering (LSR) in Technische Universität München (TUM). Firstly, I would like to thank my supervisors for considering me as a competent candidate to pursue a PhD at LSR and for providing me an opportunity to contribute towards different research projects and problems. I would also like to thank them for giving me the independence in pursuing different research ideas and providing me the state-of-the-art hardware, such as the Z+F scanner, to highlight those ideas in real world applications. During my PhD at TUM, I got the chance to attend a wide variety of scientific forums such as the Tohoku university summer school in Japan as well as flagship robotic conferences such as IROS, ICRA and ICARCV, which provided me an essential platform to discuss and share research ideas with a wide variety of researchers from the scientific community. I want to thank Prof. Dieter Fox and his PhD students from University of Washington for the lively discussions during their visit at TUM. I would like to express my gratitude to Dr.-Ing Christoph Fröhlich for inviting me to visit the Z+F premises and furthermore providing an insight into the development of the Z+F laser scanner. Finally, the research work in this thesis would not have been possible without the support of my students specifically Florian Bücherl, Athanasios Dometios, Chris Verginis, Christoph Allig and Thomas Wildgruber.

During my stay at LSR, I had the pleasure to meet as well as become friends with a lot of people whom I wish to thank from the bottom of my heart. In particular, my initial office mates Dr. Markus Rank and Dr. Stefan Klare who helped me out during the starting period of my PhD. I would like to specially thank the IURO team, i.e. Christian Landsiedel, Annemarie Turnwald, Roderick De Nijs, Nikos Mitsou and Daniel Carton, as well as other LSR team members such as Andreas Lawitzky, Daniel Althoff and Stefan Friedrich for the lively discussions and cheerful moments at the institute, especially in the coffee kitchen.

I want to thank my parents and specifically my wife Maria Iftikhar for being so supportive and helpful during the topsy-turvy time period of my PhD.

Munich, April 2016 Sheraz Khan

Abstract

The robotics research community envisions a future in which autonomous mobile robots play an important role in a multitude of real world applications such as personal health care, autonomous driving, planetary exploration as well as search and rescue operations. Recent advances in the field of intelligent and autonomous mobile robots have brought this dream closer, however, significant research challenges still remain in the domain of perception. The presence of an effective and robust perception pipeline is an essential requirement for the development of an autonomous mobile robot as it contributes towards a wide variety of robotic applications such as navigation, localization and exploration. This thesis contributes in the domain of perception by proposing novel approaches in the areas of *environment representation*, *simultaneous localization and mapping (SLAM) and loop closure detection/place recognition*. The subdomain titled *environment representation* provides the basis for creating a map of the environment by defining the geometric primitive (such as points, lines or a cubic grid) used to approximate the environment. In contrast, the subdomain of *SLAM* devises the algorithm that allows the robot to create maps in an online, incremental manner based on the geometric primitive chosen for environment representation. The final aspect of loop closure/place recognition supplies the tools for recognizing previously visited locations thereby maintaining the consistency and accuracy of the map over time by reducing the error accumulated by the SLAM algorithm. Hence, the above highlighted aspects within the domain of perception provide mobile robots with the capability of generating *accurate* and *consistent* maps of the environment in an online, incremental manner.

This thesis contributes in the domain of *environment representation* by presenting an approach that is capable of approximating the environment using a variable resolution grid. This variable resolution grid is stored in a hierarchy of axis-aligned rectangular cuboids, which is generated and adapted in an *online, incremental* fashion. The proposed approach is flexible in the sense that it allows the user to define the maximum number of children per node within the tree structure thereby effecting important characteristics such as insertion, access times as well as the number of nodes required to represent the variable resolution grid. In addition, the number of grid cells required to approximate the environment are substantially fewer in comparison to a fixed resolution grid.

Given an environment representation mechanism, another challenging aspect of the perception pipeline is the development of an algorithm that allows the robot to estimate its own pose as well as to generate an detailed map of the environment in an online, incremental manner. Hence in context of SLAM, this thesis proposes an approach that augments geometric models of the environment with a measure of surface reflectivity based on the *intensity* observations of the laser scanner. To acquire this measure of surface reflectivity a generic and simplistic calibration mechanism is presented. Furthermore, this reflectivity measure is used for simultaneously estimating the robot pose as well as acquiring a reflectivity map, i.e. occupancy grid augmented with surface reflectivity information of the environment.

An important sub-component of the SLAM algorithm is the loop closure/place recognition mechanism. This thesis contributes towards different aspects of appearance based loop closure detection/place recognition problem i.e. vocabulary generation mechanisms as well as identifying the influence of active (laser) and passive (cameras) sensors, projection models and descriptors in the performance of the algorithm. In context of vocabulary generation, an *online, incremental* mechanism for binary vocabulary generation is presented that allows appearance based loop closure detection at a high recall rate with 100% precision in comparison to the state-of-the-art algorithms. In addition, this thesis evaluates the role of different types of sensor modalities, projection models and descriptors for place recognition and furthermore highlights their advantages under challenging lighting conditions.

In summary, this thesis contributes in the areas of environment representation, SLAM and appearance based loop closure detection/place recognition within the domain of perception that allow robots to generate accurate maps of the environment in an online, incremental manner. An extensive experimental evaluation is carried out for each contribution to highlight its characteristics as well as advantages in comparison to the state-of-the-art.

Zusammenfassung

Die Forschung in der Robotik zeigt uns eine Zukunft, in der autonome Roboter im Alltag eine wichtige Rolle spielen. Beispiele dafür sind der Einsatz von Robotern in der Pflegeindustrie, dem autonomen Fahren, der Planetenerkundung und für Such- und Bergungsarbeiten. Neuste Fortschritte im Bereich der intelligenten und mobilen Robotik bringen uns diesem Zukunftstraum einen Schritt näher. Allerdings stellen sich noch bedeutende Herausforderungen im Bereich der Wahrnehmung. Eine effektive und zuverlässige Wahrnehmung der Umgebung ist eine grundlegende Voraussetzung für die Entwicklung eines mobilen Roboters, da eine Vielzahl von Anwendungen - wie die Navigation, Lokalisierung und Exploration - davon abhängt. Diese Dissertation bereichert das Forschungsgebiet der Wahrnehmung durch neue Ansätze in den Bereichen *Umgebungsmodellierung*, *Simultane Lokalisierung und Kartierung* (SLAM), sowie *Schleifenschluss* (Loop Closure) bzw. *Ortswiedererkennung* (Place Recognition). Die Umgebungsmodellierung stellt die Grundlage für das Erstellen einer Karte dar. Dabei wird die Umgebung zumeist durch geometrische Primitive - wie zum Beispiel Punkte, Linien oder kubische Grids - approximiert. Der Teilbereich SLAM beschäftigt sich dagegen mit den Algorithmen, welche es einem mobilen Roboter ermöglichen eine Karte aus den jeweils gewählten geometrischen Primitiven online und inkrementell aufzubauen. Der zuletzt genannte Bereich Schleifenschluss bzw. Ortswiedererkennung befasst sich mit dem Wiedererkennen von zuvor besuchten Stellen. Dadurch wird die Einheitlichkeit und Genauigkeit der Karten über den gesamten Zeitbereich aufrecht gehalten und eventuell auftretende Fehler der SLAM Funktion reduziert. Folglich ermöglicht eine Kombination der drei oben genannten Bereiche es einem mobilen Roboter eine *genaue* und *einheitliche* Karte der Umgebung online und inkrementell zu erstellen.

Diese Dissertation erweitert den Stand der Forschung im Bereich der *Umgebungsmodellierung* um eine Möglichkeit die Umgebung mit Hilfe eines Grids zu approximieren, dessen Rasterauflösung variabel ist. Dieser variable Grid ist hierarchisch aus axial zueinander ausgerichteten Quadern aufgebaut. Der Grid wird *online* generiert und zur Laufzeit *inkrementell* angepasst. Die Flexibilität des Ansatzes ist gewährleistet, indem der Benutzer selbst die maximale Anzahl an Kindknoten innerhalb der Baumstruktur wählen kann. Diese beeinflusst direkt die Zeiten, die nötige sind, um auf einen Knoten zuzugreifen bzw. einen neunen Knoten einzufügen. Außerdem bestimmt sie die Anzahl der Knoten, die dazu nötig sind die Umgebung darzustellen. Diese ist bei dem variablen Grid erheblich geringer als bei einem Grid mit konstanter Auflösung.

Für den Fall, dass schon ein Mechanismus zur Umgebungsmodellierung existiert, stellt sich eine neue Herausforderung für einen mobilen Roboter: das Schätzen seiner eigenen Position und das inkrementelle Erstellen einer detaillierten Karte der Umgebung zur Laufzeit. Diese Arbeit ergänzt die Forschung im Bereich SLAM durch einen Ansatz, der die Umgebung nicht nur mit geometrisch Modellen abbildet, sondern zusätzlich das Reflexionsvermögen der Oberflächen in der Umgebung mit einbezieht. Das Reflexionsvermögen basiert dabei auf den *Intensitätsmessungen* eines Laserscanners. Innerhalb dieser Arbeit wird eine generische und einfache Kalibrierungsmethode vorgestellte, die es ermöglicht das Reflexionsvermögen zu erfassen. Zudem wird gezeigt, wie die Messung des Reflexionsvermögens verwendet werden kann, um gleichzeitig die Position eines Roboters zu schätzen sowie eine auf dem Reflexionsvermögen basierende Karte aufzubauen. Diese Karte entspricht einem Besetzungsgitter, das zusätzlich Informationen über das Reflexionsvermögen der Oberflächen enthält.

Eine weitere, wichtige Maßnahme, die den SLAM Algorithmus verbessert, ist der Einsatz von Methoden für das Wiedererkennen zuvor besuchter Orte. Diese Dissertation erweitert bisherige Methoden durch folgende Aspekte: einen Ansatz zur Erzeugung von Vokabeln; und eine Untersuchung der Einflüsse von aktiven Sensoren (Laser), passiven Sensoren (Kameras), Projektionsmodellen und Deskriptoren auf die Performance des Algorithmus. Eine *online laufende* Methode zur Erzeugung eines binären Vokabulars wird vorgestellt, welche – im Gegensatz zu bisherigen Methoden – eine Detektion von Schleifenschlüssen mit einer 100% Genauigkeit (Precision) bei hoher Trefferquote (Recall) ermöglicht. Darüberhinaus wird in dieser Dissertation die Rolle von unterschiedlichen Sensor-Modalitäten, Projektionsmodellen und Deskriptoren für die Ortswiedererkennung evaluiert und deren jeweiligen Vorteile bei schwierigen Lichtverhältnissen herausgearbeitet.

Zusammenfassend lässt sich sagen, dass diese Dissertation die bisherige Wahrnehmungsforschung in den Bereichen Umgebungsmodellierung, SLAM und Detektion von Schleifenschlüssen bzw. Ortswiedererkennung erweitert. Dadurch wird es einem Roboter ermöglicht eine genaue Karte der Umgebung in einer inkrementellen und online lauffähigen Weise zu erstellen. Alle neuen Ansätze wurden umfangreich experimentell evaluiert, um deren Eigenschaften und deren Vorteile gegenüber bisherigen Ansätzen aufzuzeigen.

Contents

Notations

Abbreviations

2D	Two Dimensional
2.5D	Two and a half Dimensional
3D	Three Dimensional
BOW	Bag of Words
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
EKF	Extended Kalman Filter
EOS	Electro-Optical System
FABMAP	Fast Appearance Based Mapping
FOV	Field Of View
FPFH	Fast Point Feature Histograms
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation and Value
IBuILD	Incremental Bag of Binary Words for Appearance based Loop Closure Detection
ICP	Iterative Closest Point
ICL	Iterative Closest Line
IURO	Interactive Urban Robot
LASER	Light Amplification through Stimulated Emission of Radiation
LIDAR	Light Detection and Ranging
LUT	Look up table
MBR	Minimum Bounding Rectangle
MBRC	Minimum Bounding Rectangular Cuboid
MOCAP	Motion Capture
MROL	Multi Resolution Occupancy List
MVOG	Multi Volume Occupancy Grid
NARF	Normal Aligned Radial Features
NDT	Normal Ditribution Transform
NDT-OM	Normal Distribution Transform Occupancy Maps
RC	Rectangular Cuboid
SAD	Sum of Absolute Differences
SDF	Signed Distance Function
SFM	Structure From Motion
SHOT	Unique Signatures of Histogram for Surface and Texture description
SIFT	Scale Invariant Feature Transform

SLAM	Simultaneous Localization & Mapping
SURF	Speeded Up Robust Features
TF-IDF	Term Frequency-Inverse Document Frequency
UKF	Unscented Kalman Filter
Z+F	Zoller and Fröhlich

Symbols

General

$\mid * \mid$	Cardinality of a set
$\epsilon(*)$	Error function
$l(*)$	Log odds
$\oplus$	Motion composition operator
$\ominus$	Inverse of the motion composition operator
$P(*)$	Probability
$P(*\mid*)$	Conditional probability
$R(*)$	Reflectivity attribute of the grid
$\veebar$	Exclusive OR operator

Variables

α	Angle of incidence with respect to the surface normal
$\mathbf{d}$	Descriptor vector
$\mathbf{D}$	Matrix of descriptors
δ_{ij}	Distance between the i^{th} and j^{th} pose of the robot
δ_{ij}^{*}	Groundtruth distance between the i^{th} and j^{th} pose of the robot
η	Azimuth in spherical coordinates
g	Grid cell in a grid G
$\bar{g}$	Modified grid cell
$\mathbf{H}$	Hessian matrix
$\mathbf{I}$	Image
I_{rec}	Received intensity optical power
$\mathbf{I}^{\text{rect}}$	Image generated using the rectilinear projection model
$\mathbf{I}^{\text{eqrect}}$	Image generated using the equirectangular projection model
${}^{\text{c}}\mathbf{I}$	Color image
${}^{\text{r}}\mathbf{I}$	Range image
${}^{\text{i}}\mathbf{I}$	Intensity image
λ	Elevation in spherical coordinates
n_{g_i}	Number of observation for grid cell g_i
$\mathbf{O}$	Generic notation for a point cloud or image
p_i	i^{th} term of the polynomial
$\mathbf{P}$	Point cloud
P_{rec}	Received optical power
r	Distance or radial distance in spherical or polar coordinates

$\mathbf{R}$	Minimum and maximum bounds of a rectangle or RC
$\mathbf{r}^{\text{min}}$	Minimum bound of a rectangle or RC
$\mathbf{r}^{\text{max}}$	Maximum bound of a rectangle or RC
$\mathbf{s}_i$	Cartesian coordinates of the i^{th} observation
$\bar{\mathbf{S}}$	Search direction matrix
t	Time index
ϱ	Surface reflectivity
$\bar{\varrho}$	Relative surface reflectivity
$\mathbf{V}$	Vocabulary of local or global descriptors
x_i	x_i coordinate in an image, grid or a point in the point cloud
$\boldsymbol{\zeta}$	Robot pose
z_t	Sensor observation at time index t

Functions

centroid$(*, *, ..)$	Returns the centroid of binary valued descriptors
$H(*, *)$	Hamming distance
$\mathcal{S}(*, *)$	Similarity metric
$\bar{\mathcal{S}}(*, *)$	Normalized similarity metric

Constants

β	Temporal consistency threshold
B_δ	Ball defined by distance δ
χ	Number of scales
δ	Distance in Euclidean or Binary space
d_{apt}	Laser scanner aperture diameter
μ_{min}	Minimum probability threshold for occupancy grid update
μ_{max}	Maximum probability threshold for occupancy grid update
Ω	Orientation per scale for GIST descriptors
P_{occ}	Occupancy grid update term for occupied regions
P_{free}	Occupancy grid update term for free space
P_{emit}	Emitted optical power by the scanner
ψ	Cubic grid cell size for downsampling point clouds
σ	Window size for gaussian smoothing for HOG descriptors
τ_{sys}	Laser scanner system transmission factor
φ	Number of blocks for GIST descriptor

1 Introduction

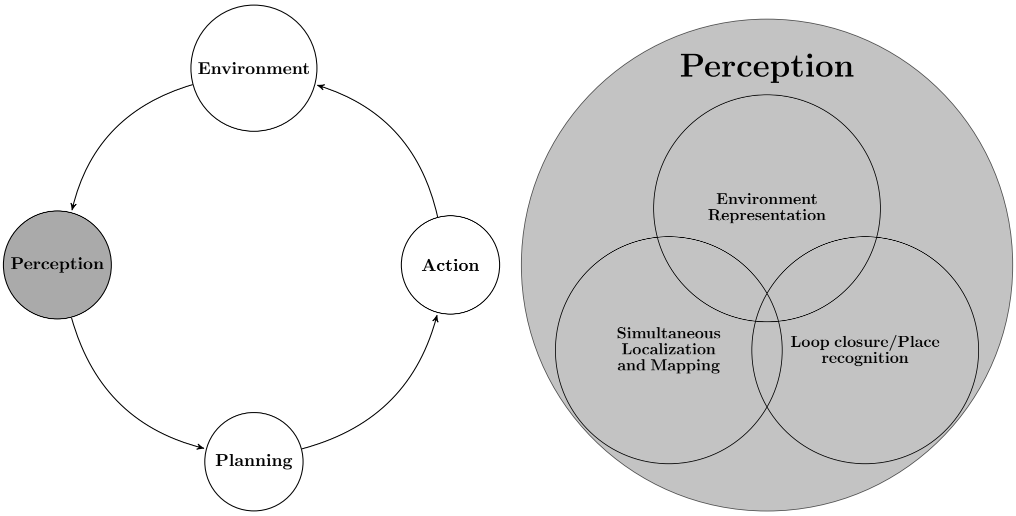
The last few decades have seen a significant amount of research in the field of Robotics, specifically intelligent and autonomous mobile robots [25, 97, 167, 183]. The main reason for this surge in interest has been the expected utility and application of the robotics technology in the domain of personal human health care, autonomous driving, search and rescue operations in disaster scenarios as well as space exploration. The focus of research within these domains differ depending on the application scenario e.g. in context of personal/assistive robots, the focal point of the research work is on social acceptance [11, 23, 34] of robots in human populated environments. In addition, the research within the robotics community has also focused on providing robots with robust perception, navigation and long term autonomy capabilities [5, 96, 209] to allow them to operate in dynamic, real world urban scenarios. One specific application of this is the Interactive Urban Robot (IURO), which aims to fill knowledge gaps via human interaction as shown in Figure 1.1 and furthermore utilize this information for achieving its goal of autonomously navigating to a certain point within the city without any map of the environment. Another interesting application of outdoor urban robotics is autonomous driving in which the Google self-driving car [1] is a well known example. In addition, different automotive companies such as BMW, Mercedes, Bosch, Uber and Tesla have also been investing heavily in research and development of driving assistance systems and fully autonomous cars. Another application of robotics technology that can have a major impact in the near future is search and rescue robots for natural disasters [91, 128, 130, 131]. A recent example of this is the Fukushima Daiichi nuclear disaster, where the main purpose of using robots was to reduce the risk of additional human casualties. In addition, autonomous robots are playing an important role in helping humans explore the frontiers of space such as the NASA Mars rover [67, 173] which is being used for planetary exploration. The examples highlighted above provide a brief glimpse into the recent research and development efforts in different applications of intelligent and autonomous mobile robots.

The application scenarios for mobile robots are quite diverse, however the core functionalities required to impart autonomous behaviour are the same across all applications. These functionalities include the capacity of perceiving the environment, planning and furthermore performing an action based on the state of the environment. The basic perception-planning-action cycle is shown in Figure 1.2(a). A typical mobile robot can have a wide range of sensors e.g. sonars, laser scanners, cameras etc. that allow it to sense the current state of the environment and furthermore it can use different perception algorithms to extract meaningful information from these sensor observations. This thesis focuses on different aspects in the domain of perception, such as *environment representation*, *Simultaneous Localization and Mapping (SLAM) and loop closure detection/place recognition*, as shown in Figure 1.2(b) using a Wenn diagram, that allow robots to generate consistent and accurate maps of the environment .

(a) Interactive Urban Robot (IURO)

(b) IURO interacting with pedestrians

Fig. 1.1: Equipping mobile robots with capabilities and functionalities that allow them to operate in real world outdoor urban environments.


Environment
Perception
Action
Planning
Perception
Environment Representation
Simultaneous Localization and Mapping
Loop closure/Place recognition

(a) Perception-Planning-Action cycle

(b) Perception

Fig. 1.2: a) The commonly used perception-planning-action cycle in the field of Robotics. b) The focus of this thesis i.e. environment representation, SLAM and Loop closure detection/place recognition highlighted as a Wenn diagram. The above mentioned research aspects are important for generating accurate and consistent maps of the environment, which is an essential requirement for a large number of applications in context of intelligent and autonomous mobile robots.

1.1 Problem Definitions & Challenges

An accurate metric or topological map of the environment is an essential requirement for a wide variety of robotic applications. The process through which a robot generates a consistent and accurate map of the environment requires certain questions need to be asked of which the following few are discussed in this thesis:

- Which geometric primitive should be used by the robot to internally approximate the environment?

- How can a mobile robot generate an detailed map of the environment based on sensor observations in an online, incremental fashion given a geometric primitive for environment representation?
- How should the robot maintain consistency of the map after revisiting a location?

The questions highlighted above inquire about the underlying concepts discussed in this thesis. The first questions is linked to the *environment representation* and inquires about the geometric primitive that should be used by a robot to approximate the complex external environment. The second question builds upon the first question by inquiring that given a specific mechanism for environment representation, how can the robot build an accurate and consistent map in an *online, incremental* manner. The algorithm that allows a robot to estimate its own pose as well as build an map of the environment is titled Simultaneous Localization and Mapping (SLAM) or Self Localization and Mapping and it has been the subject of intense research within the field of robotics in the last few decades. The final question builds upon the first two questions and focuses on maintaining the consistency of a map over time during SLAM. One specific aspect of maintaining map consistency is titled the loop closure problem in which the robot needs to determine if it is revisiting a location and furthermore use this information to reduce the uncertainty over its pose.

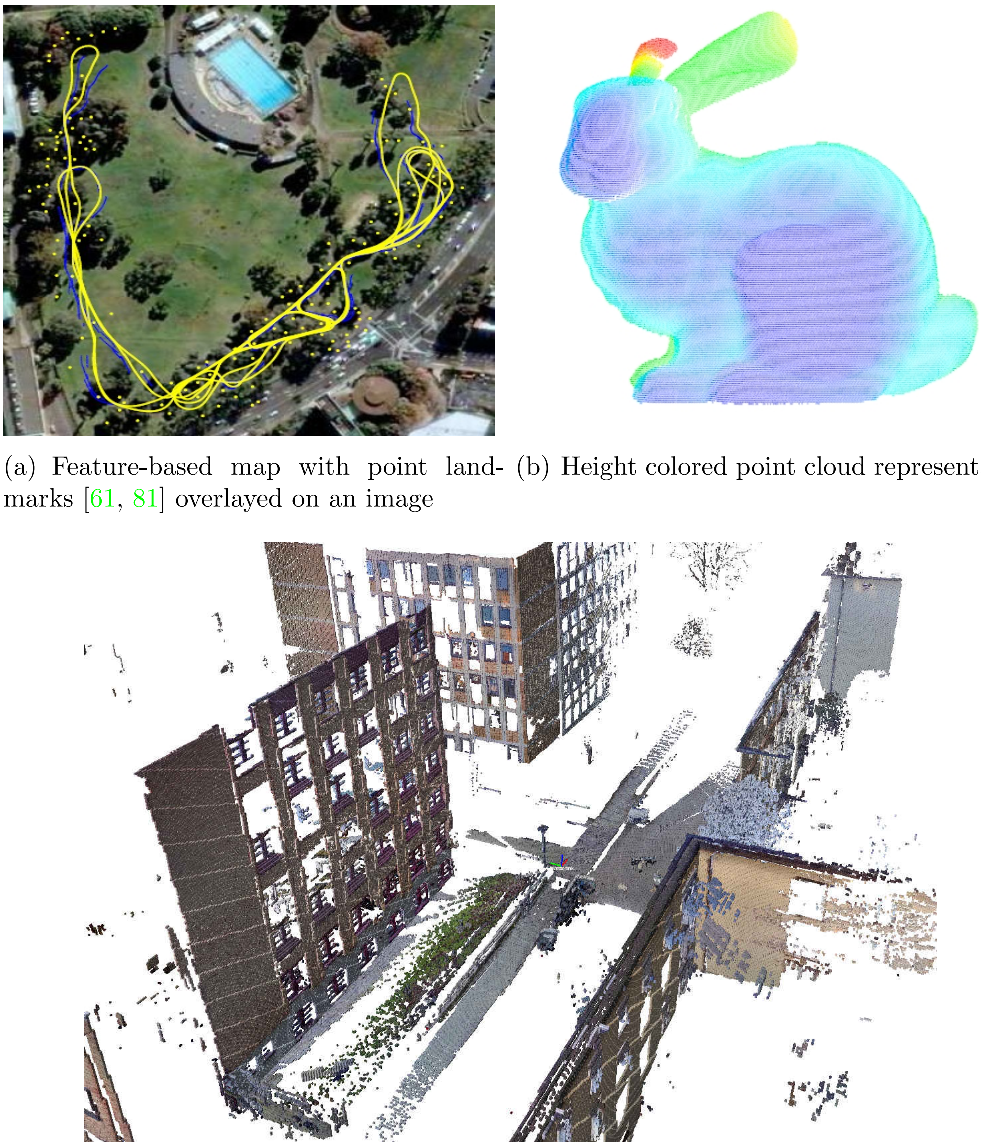
In summary, this thesis focuses on the following aspects in the domain of perception:

- Environment representation
- Simultaneous Localization and Mapping (SLAM)
- Place recognition/Loop closure detection

as shown in Figure 1.2(b), which are tightly coupled during the map creation process and play a fundamental role in providing robots the capability of generating accurate maps in an online, incremental fashion. The following subsections present an overview of the highlighted aspects.

1.1.1 Environment Representation

The *environment representation* mechanism is effectively a geometric primitive that allows the robot to generate an approximation of the external environment using sensor observations. In the computer graphics and robotics community different mechanisms have been proposed and used in literature e.g. point, surface or grid based representations. Figure 1.3(b) shows a point cloud representation in which each point represents a sample from the actual surface generated by the sensor. Figure 1.3(c) shows a grid based environment representation i.e. *occupancy grids*, which stores an occupancy probability for each grid cell. In addition, there exists landmark-based maps which (typically) approximate the environment using point landmarks. Figure 1.3(a) shows a landmark-based map in which the point landmarks are shown as yellow dots and correspond to the natural (tree trunks) and artificial landmarks (reflectors) detected in the environment whereas the robot trajectory is shown as a yellow line. The most commonly used geometric primitives for approximating the environment can be categorized as follows:



(a) Feature-based map with point landmarks [61, 81] overlayed on an image (b) Height colored point cloud representation

(c) Grid based representation (10 cm grid cell size) using the Rtree occupancy grid [89, 211] with color information

Fig. 1.3: a,b,c) Commonly used environment representations (point landmarks, pointcloud or grid based) for generating a map of the environment

- Point based representation (Point clouds and Landmark based maps)
- Surface based representation (Planes, Triangular meshes)
- 2D/3D grid based volumetric representation

The categorization above is performed to simplify the discussion, however in literature there exists no clear division due to cross coupling between primitives as one environment representation can be extracted from the others. In general, different algorithms allow extraction of a surface representation from a grid e.g. marching cubes [103] or point clouds [8, 85]. In addition there exists no standard naming convention as in robotics literature landmark (typically point based approximation) as well as line, plane based maps are also titled *feature* based maps.

Point based Representation

The most commonly used point based approximations are landmark and point cloud based representations for generating maps of the environment. Landmark based representations [81, 121] extract static, distinguishable and repeatable point observations from the robot sensor to be able to estimate the robot pose using SLAM as shown in Figure 1.3(a). In some cases these landmarks correspond to artificial markers (beacons or surfaces with high reflectivity) [62] which are manually placed in the environments. In contrast, point clouds 1.3(b) are an accumulation of 2D/3D points that represent samples from the object surface obtained from the sensor. Point cloud based representations have recently become quite popular with the advent of the Kinect, Velodyne sensors and are quite frequently used within robotic applications such as object detection, tracking and semantic labeling.

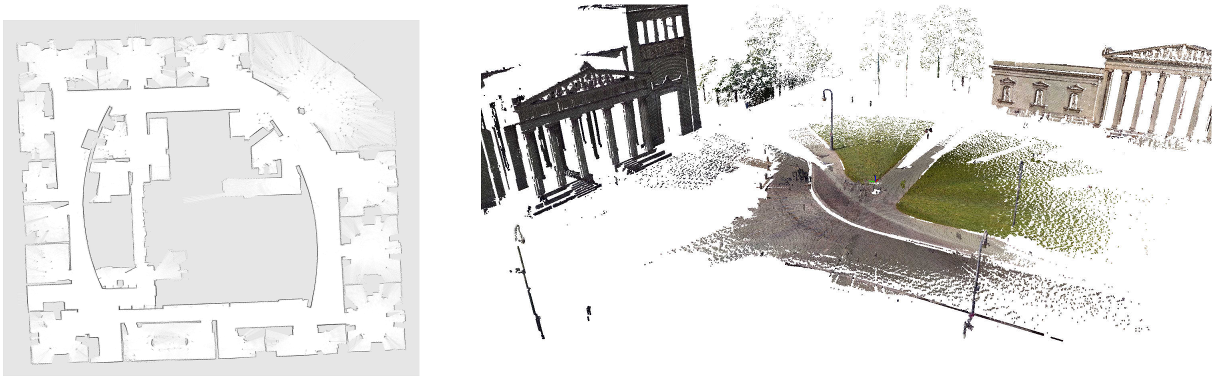
Surface based Representation

In contrast to point based representations, another approach to represent the environment is to fit lines or planes to the sensor observations leading to line [154, 207] or plane based environment maps [148, 196]. These approaches are parametric in nature as they use a specific model to represent the environment. In addition, there also exists approaches that take advantage of the orthogonality assumption in structured indoor environment (Manhattan world assumption) to place constraints between these fitted models in order to generate consistent maps of the environment [140]. Another technique that is quite popular in computer graphics/vision community [53, 54, 117] and has recently been adopted by the robotics community is the usage of triangular meshes for approximating the environment [113].

Grid based Representation

The most commonly used mechanism within the robotics community for generating maps of environment are grid based representations which discretize the environment into cells and generate a metric model of the environment. In principle, the grid can be used to store any attribute of the surface. In the domain of robotics, the most commonly used attribute is the occupancy probability which defines the probability of a specific grid cell being occupied or free and these grids are titled *occupancy grids*. In addition, there exists other approaches [136, 197] such as the truncated signed distance or the Normal distribution based representation. The truncated signed distance function (TSDF) is a signed value defining the distance of the cell to the closest surface. In contrast, the Normal distribution transform (NDT) [111] approximates the point distribution in each cell using a Normal distribution and has been used in a variety of robotic tasks such scan matching, occupancy mapping and loop closure detection.

The particular choice of an environment representation is dependent on the application, operating conditions (environment structure) as well as the sensor set available to the robot. In general, occupancy grids are the most commonly used representation as they are based on a probabilistic framework, which provides a principled mechanism for dealing with sensor noise and multi sensor fusion. The environment representation provides the basic tools for development of a map which is required in a multitude of robotic applications and essential for the development of an intelligent and autonomous mobile robot.



(a) 2D Grid based representation [64]

(b) 3D Grid based representation (only occupied regions)

Fig. 1.4: Grid based environment representations. a) 2D occupancy grid of the publicly available Intel dataset b) 3D occupancy grid augmented with color information

1.1.2 Simultaneous Localization and Mapping

Once an environment representation mechanism based on a geometric primitive has been chosen, the next step is to develop an approach that allows the mobile robot to generate a consistent, accurate map of the environment in an online, incremental manner. In the domain of robotics, such an approach or algorithm is commonly known as *Simultaneous Localization and Mapping or Self Localization and Mapping* (SLAM). The last few decades have seen a significant amount of research in the domain of SLAM, that allows a robot to simultaneously estimate its own pose as well as generate a map of the environment [58, 64, 81, 91, 121]. Figure 1.3(a) and Figure 1.4(a) shows a landmark and a 2D occupancy grid based map generated using a SLAM algorithm.

Within the robotics research community, the SLAM problem is termed as the *chicken and egg* problem because a good pose estimate is essential for determining an accurate map and vice versa. A robust solution to the SLAM problem is considered as the *holy grail* in the mobile robotics community as it allows a robot to autonomously generate a map of the environment which is essential in a wide variety of robotic applications [39]. An important characteristic common to the majority of SLAM algorithms in literature is their reliance on a probabilistic framework to deal with uncertainties i.e. noise in the applied control input (motion update) and the sensor observations. In general, SLAM approaches can be classified into two different categories i.e. filtering or smoothing algorithms. Typical filtering based SLAM approaches are based on landmark or grid based environment representation and commonly use the extended Kalman [62, 99] or particle filter [58, 121] to estimate the robot pose as well as the landmark positions. Recently, smoothing based SLAM approaches have become quite popular as they allow a principled mechanism for incorporating loop closure constraints thereby considering previous states in the estimation process which are forgotten in a filtering based approach due to restrictive assumptions. The majority of smoothing approaches rely on a graph based representation and furthermore used nonlinear optimization techniques for estimating the complete robot trajectory [57, 81, 95]. In literature, graph based SLAM approaches are composed of two main components: the front-end and the back-end. The front-end deals with raw sensor data to estimate the robot pose and generates a graph that defines the robot trajectory by incrementally adding constraints between consecutive robot poses. In addition, the front-end also generates loop closure constraints i.e. when the robot returns to previously visited location after a long time interval. Given the consecutive robot pose as well as the loop closure constraints, the back-end estimates the posterior distribution over the complete robot trajectory.

As mentioned earlier, a key component of the SLAM front-end is the transformation estimation process between consecutive robot poses. In literature there exist *simple* incremental pose estimation techniques titled *scan matching* approaches which are sufficient for generating a map of the environment in specific cases when the robot does not encounter loop closure constraints and the mapped environment is small as discussed in different papers [72, 90]. The most commonly used approach for scan matching is the typical Iterative Closest Point (ICP) [9] algorithm. Different variants of the standard ICP algorithm [142, 157] have been proposed in literature that improve upon different aspects of the original algorithm such as computational complexity by performing nearest neighbor assignment using a Kdtree [56, 142]. In [107] different outlier rejection mechanisms for correspondence estimation are presented whereas the approach in [24] proposes a different metric, i.e the point to plane metric, for estimating the transformation between point clouds. In the category of point to point metric, there exist approaches that operate in a different coordinate system such as polar coordinates leading to polar scan matching [37]. Another variant of the standard ICP is the Iterative Closest Lines (ICL) [102] algorithm that matches lines between consecutive scans to estimate the robot pose. In addition to the techniques mentioned above that operate on a point cloud or features, certain approaches formulate the pose estimation process on a grid based environment representation. An example of this is the Hector SLAM approach [90] that frames the pose estimation process over an occupancy grids and furthermore uses the Gauss-Newton optimization to align the laser scanner observations with an already created map. The proposed approach is capable of using gradient based methods in a nonlinear optimization by performing bilinear interpolation on the occupancy grid. In contrast, the Normal Distribution Transform (NDT) [10, 110, 111] stores a Normal distribution defining the point distribution in each grid cell and furthermore frames the pose estimation process using Newton's optimization.

In addition to the characteristics of the SLAM algorithm, another aspect is related to the environment representation used by the algorithm. A large amount of research work in SLAM focuses on feature based scan matching [148] or SLAM [39, 50, 120, 122, 195]. In mobile robotics community feature based SLAM mainly consists of point based [39, 72, 120, 122] or surface based environment representations [50, 148, 148, 195]. Another approach is to utilize a grid based environment representation among which the most commonly used approach is the occupancy grid [58, 59, 90]. In addition, there exist alternatives such as the

Normal Distribution Transform [10, 110, 111] or the signed distance function (SDF) [32] which has recently been made popular by the Kinect fusion [136] and Kintinous [197] approach.

A large amount of research work has focused on different characteristics of the SLAM algorithm i.e. the pose estimation problem as well as the mapping process using different environment representations. The development of a robust SLAM algorithm is essential for creating a consistent and accurate environment map. In addition, these maps are an essential requirement for the development of the wide range of functionalities for an intelligent and autonomous robotic system.

1.1.3 Place Recognition/Loop Closure Detection

A key component of the SLAM algorithm is the place recognition/loop closure mechanism that allows the robot to maintain a consistent map of the environment over time after a robot revisits a location. The objective of the place recognition/loop closure mechanism is to determine if a specific sensor observation (an image or point cloud) has been previously observed in a metric map or a database using a similarity metric. The place recognition problem originates from the field of computer vision specifically in the domain of content based image retrieval from databases [60, 170]. A specific instance of the place recognition problem titled the *loop closure* problem is commonly discussed in robotics literature. Loop closure is considered as a sub-problem of place recognition due to the presence of additional constraints such as the temporal consistency constraint over sensor observations or the presence of odometry (motion estimates). A robust solution to the loop closure problem in the field of robotics is an essential requirement for maintaining the consistency and accuracy of the geometric or topological map over time. Figure 1.5 shows a simple example in context of laser based SLAM, which is equally applicable for other sensor modalities as well, in which a robot is unable to determine if it has returned to a previous location and therefore the accumulated error in the pose estimates leads to an inconsistent map.

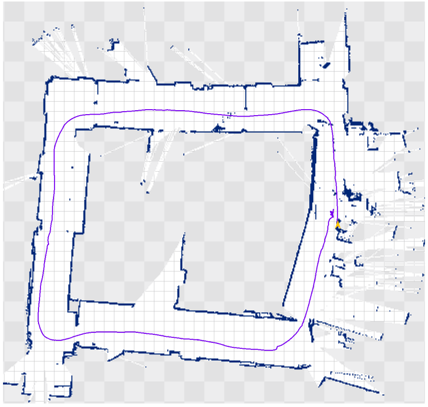


Fig. 1.5: The inability of the algorithm in detecting the loop closure constraint in context of laser based SLAM leads to an inconsistent metric map of the environment.

The problem of loop closure has been addressed in literature from different perspectives depending on the type of sensor modalities used by the robot. Typically, laser based loop closure mechanisms rely on geometric information [16, 55, 175, 176, 208], whereas in the last decade with the advent of information rich sensors such as cameras and high-end terrestrial laser scanners as well as the increase in computational power, the research focus has shifted towards appearance based mechanisms [3, 31, 116, 118] or approaches that combine metric and appearance information [69, 149, 214]. Appearance based mechanisms can be roughly classified into *local* and *global* descriptor based approaches. *Local* descriptor based approaches extract highly discriminative keypoints in an image and furthermore generate a compressed description of the region around those keypoints. Furthermore, these descriptors are typically used in a bag of words approach [3, 31, 49, 52, 141, 213] to detect loop closures or recognize places. In contrast, *global* [118, 127, 168, 179] descriptors summarize the complete image in order to recognize similar locations. An aspect common to both approaches is the requirement of a suitable metric to quantify the similarity between images. In the domain of loop closure, most approaches take advantage of the temporal consistency over sensor observations as the robot traverses the environment. The removal of the temporal consistency constraint, odometry, and GPS information transforms the loop closure problem into the standard place recognition problem addressed within the computer vision community in which images corresponding to the query image are retrieved from a database based on a similarity metric.

The main challenges being faced by loop closure/place recognition algorithms in real world robotic applications can be classified as intrinsic or extrinsic. Extrinsic challenges occur due to variations in the structure of the environment. The main extrinsic challenge for place recognition algorithms operating operating under challenging lighting conditions with passive sensors (such as cameras) in typical outdoor scenarios is the change in the environment appearance due to variations in ambient lighting (transition from day to night time). Even during different times of the day, shadows can cause a change in the environment appearance and pose challenges for place recognition algorithms [108, 115, 118]. In contrast to the extrinsic challenges mentioned above, intrinsic challenges correspond to deficiency of prior information available to the algorithm such as the lack of motion estimates (odometry) or the unavailability of GPS. In addition, intrinsic challenges might also include the deficiency of prior training data for generating a visual vocabulary which is typically the case in online robotic and computer vision applications as it is assumed that no prior information is available about the environment. The extrinsic and intrinsic aspects mentioned above form a substantial set of challenges faced by place recognition algorithms in the field of robotics as well as computer vision. The development of a robust place recognition algorithm capable of addressing the above mentioned challenges is essential for the development of a robust SLAM algorithm as well as developing consistent maps of the environment over a long period of time.

1.2 Thesis Contributions

This thesis contributes in the domain of perception specifically environment representation, SLAM and place recognition/loop closure detection. The above mentioned aspects play a critical role in the development of an accurate and consistent map of the environment. These maps are essential for different robotic applications such as navigation and exploration and play a fundamental role in the development of an intelligent and autonomous robotic systems. The following subsections describe the contribution of this thesis in the highlighted areas.

1.2.1 Environment Representation

A major contribution of this thesis is in the domain of grid based environment representation. This thesis presents an approach which is capable of approximating the environment based on a variable resolution grid in an *online, incremental* manner. The following aspects are important in defining a grid based environment representation

- Spatial decomposition
- Attribute used to represent the surface

The spatial decomposition defines the structural properties of the grid e.g. the resolution of cells and specific assumptions about their shape. In contrast, the second aspect defined above corresponds to the attribute used to represent the surface e.g. occupancy probabilities [41], Normal distribution [10, 110, 111] or the signed distance function [32].

This thesis contributes in the domain of environment representation by defining an interplay between the spatial decomposition of the occupancy grid as well as the surface attribute. In context of spatial decomposition this thesis proposes an approach that *relaxes the cubic grid cell assumption common to most occupancy grids to allow an approximation of the environment using a variable resolution grid based on a hierarchy of axis aligned rectangular cuboids (3D)*. The proposed approach allows the user to define the maximum number of children per node within the hierarchy thereby influencing the height, width of the tree and consequently effecting the insertion, access time as well as the number of nodes required in the hierarchy to represent the environment. In context of the attribute used to represent the surface, a simplistic fusion mechanism based on occupancy probabilities is presented that merges neighboring grid cells to generate variable resolution grid cells. The main motivation for using rectangular cuboids instead of cubes is the fact that they are better capable of approximating typical indoor and outdoor urban environments consisting of walls and flat surfaces.

In summary, the main contributions of this thesis in context of *environment representation* are as follow

- An approach capable of modeling the environment using a variable resolution grid (Section 2.4 and 2.5.1)
- A simplistic fusion process that couples the surface attribute i.e. occupancy probability with the spatial decomposition leading to variable resolution representations of the environment in an *online, incremental* fashion (Section 2.5.2)
- An extensive experimental evaluation highlighting the characteristics of the proposed approach on a publicly available dataset (Section 2.6)

1.2.2 Laser Intensities for SLAM

The majority of the research work carried out in the domain of SLAM focuses on using sensor observations obtained from a laser scanner to generate a consistent and accurate geometric representation of the environment. In addition to measuring the distance, a typical laser scanner also quantifies the received optical power after reflection from the object titled *intensity*. The important aspect of laser intensities is that they are dependent on an intrinsic surface property i.e. surface reflectivity as well as other extrinsic parameters such as distance and angle of incidence to the surface. Hence, it should be possible to model the influence of extrinsic parameters in order to acquire a measure of surface reflectivity.

The main contribution of this thesis is a simplistic calibration mechanism for laser scanners to acquire a *pose-invariant* measure of surface reflectivity. In addition, this measure of surface reflectivity is used in a SLAM algorithm (Hector SLAM) to simultaneously estimate the robot pose and acquire a reflectivity map of the environment. The capability of acquiring a measure of surface reflectivity provides the possibility of using this information in a variety of robotic application such as global localization, navigation and exploration. Specifically speaking reflectivity maps can be useful in scenarios where geometric information is ambiguous e.g. a symmetric corridor. It is important to define the scope of the proposed approach within the SLAM literature. The approach proposed in this thesis serves as a component of the SLAM front-end as it determines the constraints between consecutive robot poses and furthermore generates a reflectivity map of the environment.

In summary, the contribution of this thesis in context of *SLAM* is mentioned below

- A simple calibration process for laser scanners to acquire a pose-invariant measure of surface reflectivity (Section 3.3.2)
- An extension of the Hector SLAM algorithm that relies on a measure of surface reflectivity for simultaneously estimating the robot pose and acquiring a reflectivity map of the environment (Section 3.4)
- An extensive experimental evaluation of the proposed calibration approach and the Hector SLAM extension (Section 3.5)

1.2.3 Place recognition/Loop closure detection

The thesis contributes towards two different aspects of the loop closure/place recognition problem. Firstly, it focuses on the issue of vocabulary generation and proposes an approach that is capable of generating a binary bag of words (BOW) vocabulary in an *online, incremental* manner for online robotic applications. Secondly, this thesis evaluates the advantages of using laser intensities for the place recognition problem under challenging lighting conditions. The following paragraphs provide a detailed perspective on the contributions of this thesis.

The BOW approach is the most prevalent approach for loop closure detection/place recognition and image retrieval in the robotics and computer vision community [141]. In context of online robotic applications such as SLAM it is assumed that the robot has no prior information about the environment, so it is considered *desirable* that the loop closure mechanism is capable of operating and in an online, incremental manner without requiring any offline processing. This thesis contributes a simplistic mechanism for generating a binary vocabulary in an online, incremental manner. Although online vocabulary generation mechanisms exist for real valued descriptors, however the typical Euclidean distance as well as clustering mechanism e.g. Kmeans are no longer applicable in binary spaces. The main advantage of using binary vocabularies based on binary descriptors is that in comparison to real valued descriptors they are less expensive in terms of computation and memory cost [100]. The proposed approach couples the vocabulary generation mechanism with a simplistic similarity metric and temporal consistency constraint to show that it is capable of generating high precision, recall in comparison to the state of the art.

In addition, this thesis evaluates the performance of different modalities under challenging lighting conditions as this is an essential stepping stone for long term autonomy in outdoor urban environments. The majority of the research work in this domain focuses on using passive sensors i.e. cameras to propose algorithms that are capable of dealing with ambient lighting conditions. In contrast this thesis focuses on active sensors i.e. laser scanners and specifically the usage of laser intensities for appearance based loop closure/place recognition. The main advantage of active sensors is their invariance to external lighting conditions. Hence, the contribution of this thesis is to highlight the advantage and applicability of laser intensities for appearance based place recognition under challenging lighting conditions in comparison to images from camera's (passive sensor) and laser scanner based geometry information.

In summary, the main contribution of this thesis in context of *loop closure/place recognition* are

- An *online, incremental* mechanism for binary vocabulary generation for loop closure detection (Section 4.4)
- To highlight the applicability and advantages of laser intensities for place recognition under challenging lighting conditions in comparison to other forms of sensor data such as images from camera's (passive sensor) or geometry information from laser scanner (Section 4.5)
- An extensive experimental evaluation highlighting the advantages of the proposed binary vocabulary generation mechanism and laser intensities in the loop closure/place recognition pipeline on real world datasets (Section 4.6)

1.3 Outline of Thesis

The outline of this thesis follows the steps required in the perception pipeline to build a consistent and accurate map of the environment i.e. the environment representation, SLAM and finally loop closure/place recognition detection. The above mentioned aspects are tightly coupled during the map creation process. The environment representation provides the basis for map generation by defining the geometric primitive used to approximate the environment. The domain of SLAM uses the geometric primitive chosen for environment representation and couples it with the pose estimation process to allow the robot to incrementally generate the map based on sensor observations. Finally, the loop closure detection/place recognition algorithm provides the capability of maintaining the consistency of the map over time by associating previously visited locations and reducing the drift accumulated by the SLAM algorithm due to motion and sensor uncertainty.

Chapter 2 focuses on environment representations and presents the details of the proposed variable resolution occupancy grid based on a hierarchy of axis aligned rectangular cuboids. This chapter highlights the key characteristics of the proposed approach using different sensor models and presents an extensive experimental evaluation in comparison to the state-of-the-art Octomap approach on a publicly available dataset. Finally, the conclusion and future work is highlighted for the proposed approach.

Chapter 3 proposes an approach that uses laser intensities in context of Simultaneous Localization and Mapping (SLAM) to acquire a reflectivity map of the environment. The chapter begins by explaining a simple calibration process for acquiring a pose-invariant measure of surface reflectivity. This measure is furthermore used in an extension of Hector SLAM that allows the robot to simultaneously estimate its own pose as well as acquire a geometric occupancy grid model of the environment augmented with surface reflectivity information i.e. reflectivity map. An extensive evaluation is carried out to highlight the pose estimation accuracy of the proposed approach as well as the advantage of generating reflectivity maps of the environment using different laser scanners.

Chapter 4 discusses two different aspects of the loop closure/place recognition problem: firstly a simplistic *online, incremental* mechanism for binary vocabularies generation. An extensive experimental evaluation in terms of precision-recall on publicly available dataset is carried out to highlight the advantages of the proposed binary vocabulary generation approach in comparison to the state-of-the-art. Secondly this chapter highlights the applicability and advantages of laser intensities for loop closure/place recognition algorithms under adverse lighting conditions. An extensive experimental evaluation using different modalities, projection models and descriptor characteristics is carried out to highlight the relevance of laser intensities for place recognition.

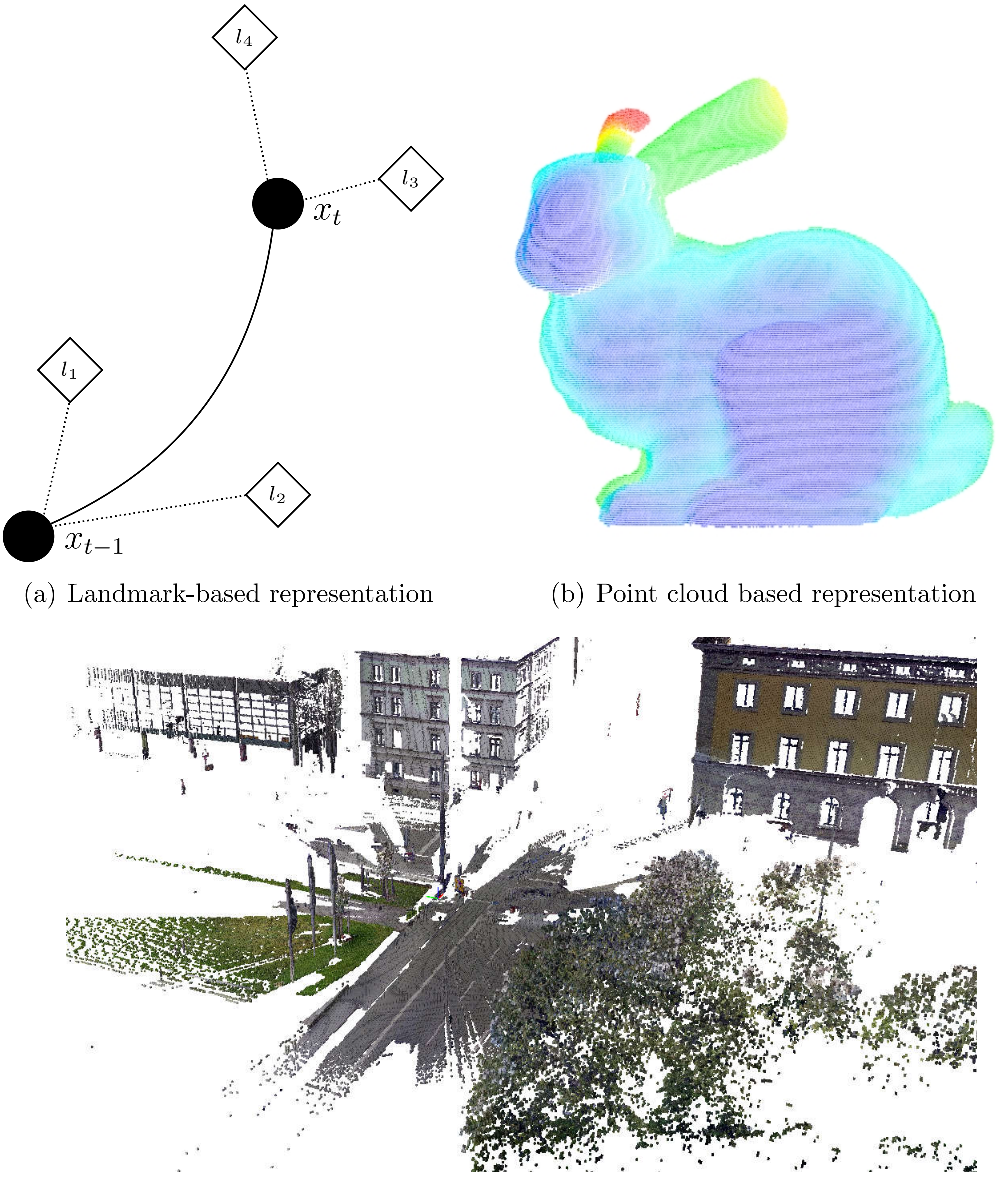
Chapter 5 summarizes the contribution of this thesis and furthermore highlights possible future research directions.

2 Environment Representation

Summary and Contribution: *This chapter presents a contribution in the domain of grid based mapping by proposing a spatial decomposition approach that is capable of modeling the environment using a variable resolution grid. This grid is stored in a hierarchy of axis-aligned rectangular cuboids that can be adapted in an online, incremental manner. Hence, the proposed spatial decomposition relaxes the cubic grid cell assumption common to a majority of grid based environment representations to allow axis-aligned rectangular cuboids. Furthermore, an extension of the standard occupancy grid is presented that couples the spatial decomposition with the attribute used for surface representation i.e. occupancy probability. This coupling is performed by adding a fusion process based on occupancy probabilities that adapts the resolution of the grid cells in an online, incremental manner, thereby generating variable resolution grid based environment approximations. An extensive experimental evaluation is carried out on a publicly available dataset using different sensor models to highlight the characteristics of the proposed approach.*

2.1 Introduction

An *accurate* environment map is an essential requirement for a large number of robotic applications such as navigation and exploration. In order to build a map of the environment, an autonomous agent requires a mechanism to approximate the complex external environment observed through its sensors. This mechanism titled the *environment representation* is essentially a geometric primitive, which is used to generate a model of the environment based on sensor observations. Hence, the environment representation can be considered as the core foundation required to build a map. Typical examples of environment representations include point based approximations i.e. point clouds, landmark-based maps [62, 121] or grid based approximations [41, 73, 136, 181, 197]. Generally landmark-based maps represent the environment using point landmarks [62, 121] which are used by the robot for pose estimation. In contrast, point clouds and grid based approximations lead to metric maps that capture the complete area or volume. Figure 2.1(a) shows a landmark-based map of the environment, which is composed of nodes that represent the robot positions in the environment and the edges corresponds to the distances between robot poses as well as landmarks detected in the environment. In contrast, Figure 2.1(b) and 2.1(c) show a point cloud and grid based metric environment representation. In principle, each environment representation has its own advantages/disadvantages and the preference of one representation over the other is dependent on a variety of factors, which include the specific application being considered as well as computational and memory constraints.



(a) Landmark-based representation

(b) Point cloud based representation

(c) Grid based representation

Fig. 2.1: Different types of environment representations. a) A landmark-based map in which static, distinguishable, repeatable point observations (l_i) are used to represent the environment. b) Point cloud based representation in which each point is a sample from the surface of the object. c) An occupancy grid based representation augmented with color information.

This chapter focuses on grid based environment mapping. A typical grid based representation has different characteristics which define its nature such as

- Spatial decomposition
- Attribute used for representing the surface

In the field of robotics the most commonly used spatial decomposition is a fixed resolution grid with cubic grid cells. In addition to the spatial decomposition, another aspect of the grid is the attribute used to store information about the surface e.g. occupancy probability, Normal distribution or the signed distance function. Grid based representations using occupancy probabilities are titled *occupancy grids* and assigns to each cell a binary random variable that defines the probability of it being occupied. This occupancy probability can furthermore be thresholded to obtain different occupancy states such as occupied (high occupancy probability), free (low occupancy probability) and unknown cells (no sensor observations). Occupancy grids are among the most commonly used approaches for navigation [41, 129, 182], exploration [17, 174, 203] as well as multi-sensor fusion [94, 114] in the domain of robotics. The main reason for the popularity of occupancy grids is their probabilistic nature, which provides a principled mechanism for dealing with multisensor fusion as well as sensor noise.

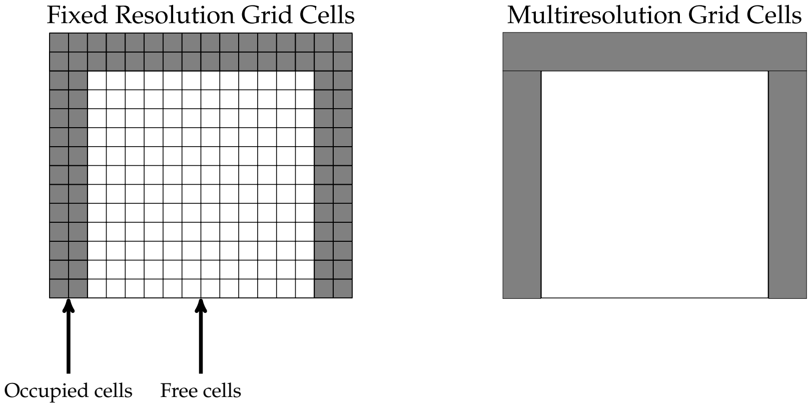


Fig. 2.2: Comparison of a fixed and a variable resolution grid representation. The variable resolution grid representation requires fewer number of grid cells in contrast to a fixed resolution representation.

In contrast to occupancy grids, the signed distance function (SDF) or truncated signed distance function (TSDF) stores a signed value in each grid cell that defines the distance to the surface thereby simplifying the process of surface extraction. In principle, the signed distance function originated from the computer graphics community [32], however it has become popular in the field of robotics with the advent of Kinect fusion [136, 197]. This chapter focuses on occupancy grids and presents a coupling of the *spatial decomposition* and *occupancy probabilities* (i.e. attribute used to represent the surface) which allows approximation of the environment using a variable resolution grid. This reason for focusing on occupancy grids is due to their popularity and wide spread usage in the field of mobile robotics.

During the last few decades the majority of the research work in the field of robotic mapping has focused on generating 2D grid based environment representations [183, 184]. Although 2D maps are sufficient in planar environments, however this assumption does not hold in a variety of indoor and outdoor environments. Recently, with the advent of higher computational power as well as advances in sensor technology such as the Kinect or Velodyne, the focus in the robotics research community has shifted towards *large scale* 3D mapping. The majority of occupancy grid based approaches in literature constitute of fixed resolution cubic grid cells. Figure 2.2 shows a fixed resolution representation in comparison to a variable resolution grid in a simplified 2D example. Intuitively speaking, this leads to a reduction in the number of required grid cells without any loss of information in the environment representation. Additionally, it allows faster access times as less number of grid cells need to be accessed to reconstruct the environment in contrast to a fixed resolution representation. If the structure (occupied regions) of the actual 3D world is composed of planar axis-aligned surfaces whereas free space does not have any definite shape, the question arises if there is any advantage in relaxing the assumption of 3D representation based on cubes (inherent to most occupancy grids) to allow axis-aligned rectangular cuboids. The objective of this chapter is to propose a variable resolution grid based environment representation and highlight its characteristics as well as advantages.

2.2 Related Work

2D occupancy grids [41, 125, 181] are considered as the de facto standard for mobile robotic mapping. Although 2D maps are sufficient in planar environments, however this assumption does not hold in a variety of indoor and outdoor environments. To deal with such scenarios different approaches have been proposed in literature such as 2.5D occupancy grids. A typical example of 2.5D occupancy grid is an elevation map [68] which stores a height value for each cell on a 2D grid. In [191], an extension titled multi-level surface maps has been proposed which allows storage of multiple heights per cell. In general 2.5D occupancy grids are useful for mapping, localization and navigation, however they are unable to model the explicit shape of the environment. In [38], an extension of multi level surface maps titled Multi Volume Occupancy Grids (MVOG) is presented which generates 3D maps by storing positive readings (observations corresponding to objects) as well as negative reading (free space readings) in vertical volumes over a 2D occupancy grid.

The recent advances in the domain of sensor technology has shifted the focus of the robotics research community from 2D towards 3D environment representations. Grid based or volumetric representations (specifically occupancy grids) and raw point clouds are the most commonly used approaches for 3D environment representations. There also exists surface based representations that extract triangular meshes or fit planes to the point cloud, however these approaches do not explicitly model free or unknown regions which is essential for a variety of mobile robot applications. Similarly, point cloud representation do not model free or unknown regions and also do not allow probabilistic data fusion from multiple sensors. One possible approach to model the environments using 3D grids is to use a dense 3D array [124, 156], however this approach is quite memory expensive due to the presence of large amount of free space in typical indoor/outdoor environments thereby limiting their usability [38, 73] for large scale mapping. In contrast, hash table based 3D representations are also used due to the amortized constant lookup time. In the field of 3D robotic mapping, MROL [161] is an approach that uses voxel lists to store occupied cells using hash tables with the keys being the closest integer grid indexes. In [160], a counting bloom filter with different hashing functions is proposed to stored occupied grid cells and the authors claim that the lookups operations can be performed within 10% of the time required for dense 3D arrays.

In contrast to the approaches mentioned above, a tree based representation for modeling a grid is also a commonly used approach within the robotics and computer vision community. Typical examples of such structures include Quadtrees for 2D [79, 202, 205] and Octrees for 3D mapping. A large amount of research has been carried out on the usage of Octrees for 3D mapping [43, 45, 147, 150]. Recently, a fully probabilistic 3D occupancy grid using octrees titled *Octomap* has been proposed which allows multiresolution 3D environment representations [73, 200]. In computer graphics literature there exists an extension of the Octree structure titled N^3 tree which allows each dimension to be divided by any arbitrary number N [28, 98]. The authors in [40] presents an N^d-tree based formulation which allows to split any d dimensional space by an arbitrary number N. The N^d tree based approach adapts the resolution of the grid in an online, incremental manner based on sensor observations. In [10, 111], the authors present an approach that stores the point distribution in each grid cell using a Normal distribution. The proposed approach uses the point distribution in each cell to estimate the robot pose using an optimization based on the Normal distribution transform (NDT). Recently, the 3D NDT (Normal Distribution Transform) [111] has been applied in the context of occupancy mapping titled NDT-OM (Occupancy Mapping) [162, 163] as well as localization in dynamic environments [193].

2.3 Contribution

This thesis contributes in the domain of grid based mapping by proposing a spatial decomposition approach that that is capable of modeling an environment using a variable resolution grid. This capability relaxes the fixed resolution cubic grid cell assumption common to most occupancy grids. The proposed approach stores the variable resolution environment approximation in the Rtree data structure [63, 133] which is composed of a hierarchy of axis-aligned rectangular cuboids. The approach presented in this chapter allows online, incremental generation and adaptation of the grid as well as the tree hierarchy based on sensor observations, which is desirable for robotic applications. In addition, the proposed approach allows the possibility of defining the maximum number of children per node in the hierarchy thereby influencing the height and width of the tree and indirectly affecting the insertion and access times of the grid cells. An extensive evaluation is carried out in this chapter to highlight the advantages of the proposed *spatial decomposition* approach. The main characteristics of the proposed spatial decomposition approach are

- *Incremental*: Allows incremental generation and update of the grid structure and the hierarchy based on sensor observations
- *Flexible*: Provides the flexibility of selecting the maximum number of children per node
- *Multiresolution grid cells*: Capable of modeling a variable resolution grid

In addition to the spatial decomposition approach, this chapter presents an extension of the standard occupancy grid by proposing a fusion process which incrementally adapts the resolution of the grid cells based on occupancy probabilities. Hence, this fusion process couples the spatial decomposition with the attribute used to represent the surface i.e.

occupancy probability. An evaluation of the proposed approach is carried out on a large scale outdoor urban dataset to highlight its characteristics and advantages in comparison to the state of the art Octomap approach.

In summary, the main contributions of this thesis in context of *environment representation* are as follow

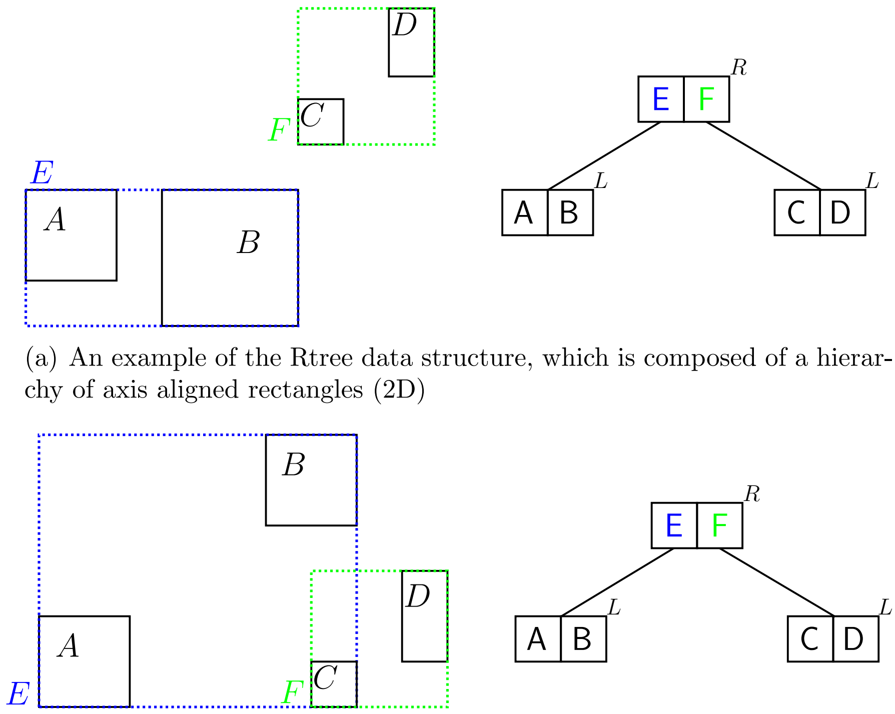
- An approach capable of modeling the environment using a variable resolution grid (Section 2.4 and 2.5.1)
- A simplistic fusion process that couples the surface attribute i.e. occupancy probability with the spatial decomposition leading to variable resolution representations of the environment in an *online, incremental* fashion (Section 2.5.2)
- An extensive experimental evaluation highlighting the characteristics of the proposed approach on a publicly available dataset (Section 2.6)

2.4 Rtree Data Structure

This section provides a brief overview of the Rtree datastructure [63, 133, 211], which forms the basis of the variable resolution occupancy grid proposed in this chapter. The Rtree structure [63] is a spatial indexing method proposed by Antonin Guttman and developed for applications within the database community. The structure is composed of a hierarchy of minimum bounding axis aligned rectangles (MBR), or minimum bounding axis aligned rectangular cuboids (MBRC) for 3D, as shown in Figure 2.3. The Rtree nodes are labeled on the upper right corners with R, L to denote root and leaf nodes respectively. Inner nodes are not shown in the Figure 2.3, however as the hierarchy expands, inner nodes are added as well. The root and inner nodes contain the MBR of the rectangles or rectangular cuboids (3D) stored in the leaf nodes. As shown in Figure 2.3, it is possible that the MBR in the Rtree hierarchy overlap. The Rtree of order $(d{,}M)$ has the following basic characteristics [63], [133]:

- A leaf node can store a maximum of M entries and a minimum of d, where $d \leq \frac{M}{2}$. The i^{th} entry in the leaf node contains the tuple $(\mathbf{R}_i \quad o_i)$. $\mathbf{R}_i$ represents the minimum and maximum bound of the 2D rectangle or 3D rectangular cuboid whereas o_i represents an attribute of this bounding rectangle e.g. occupancy probability or signed distance function etc. As the Rtree is height balanced, all leaf nodes are at the same height.
- An inner node contains a maximum of M and a minimum of d entries. Each entry consists of a MBR and a pointer to its child node.
- The root node can have a minimum of two entries unless it is a leaf node.

The following subsection explains the set of operations that can be performed on the hierarchy as well as the characteristics of the Rtree structure. In addition, an example for a simplistic 2D case is provided to highlight the Rtree hierarchy construction process.



(a) An example of the Rtree data structure, which is composed of a hierarchy of axis aligned rectangles (2D)

(b) An example of the Rtree hierarchy with overlapping MBR nodes

Fig. 2.3: An example construction of the Rtree hierarchy with maximum 2 children per node

2.4.1 Operations on the Rtree hierarchy

The Rtree structure allows operations such as

- Search
- Insertion
- Deletion

of rectangles or rectangular cuboids in its hierarchy. The rest of this section highlights the details of these operations.

Search

A search can be carried out in the Rtree hierarchy using a wide variety of criterion such as overlap, containment, intersection etc. Given a query rectangle and a criterion e.g. overlap, overlap tests can be carried out throughout the hierarchy starting from the root node and traversing onwards. The MBR or MBRC that do not overlap with the query rectangle are pruned out in the search process. If the MBR or MBRC of the inner nodes overlap, it is possible that multiple subtrees, i.e. child nodes, might need to be traversed. The search process is carried out till all overlapping entries in the leaf nodes have been tested to find if any rectangle or RC satisfies the search criterion.

Insertion

The Rtree structure hierarchy generation process is based on the insertion operation. A rectangle is inserted into the Rtree hierarchy using the *least expansion* principle in which the Rtree hierarchy is traversed by choosing an entry in the node whose MBR requires minimum expansion. In case of ties the rectangle with the smaller area is chosen. This traversal continues until a leaf node is reached, in which that rectangle is inserted. During the insertion process if the number of entries/elements in a node exceed the maximum number M, the node overflows and has to be split. An ideal splitting algorithm would distribute the entries of a splitting node between two nodes in a way that the corresponding MBR/MBRC of the entries in the parent nodes would have no overlapping area thereby reducing the chance that both these nodes will be traversed during a subsequent search. In literature there exist different splitting strategies such as linear, quadratic, exhaustive [63] or Rstar [7]. In this thesis the quadratic splitting algorithm is used as it provides a reasonable trade off between the computational complexity of exhaustive search and the worse quality of splits in case of the linear splitting algorithm. The following subsection provides details about the quadratic splitting algorithm.

Quadratic Splitting

The quadratic splitting algorithm [63] distributes the entries of a node between two nodes using the area of the rectangles or MBR as a criterion. The quadratic splitting algorithm consists of two important aspects, firstly the process to choose the *seeds* i.e. the pair of rectangles/RC/MBR/MBRC which form the first entry of the two nodes. Secondly, the process of incrementally assigning the rest of the rectangles/RC/MBR/MBRC to those nodes.

The quadratic splitting algorithm chooses the seeds by iterating over all pair of rectangles/RC/MBR/MBRC and calculates their individual area e.g. $a_{\mathbf{R}_i}, a_{\mathbf{R}_j}$ and their composition $a_{\mathbf{R}_{ij}}$ and then calculates the metric $J(a_{\mathbf{R}_i}, a_{\mathbf{R}_j}) = a_{\mathbf{R}_{ij}} - a_{\mathbf{R}_i} - a_{\mathbf{R}_j}$. The algorithm chooses the pair which leads to the largest value of the metric $J(a_{\mathbf{R}_i}, a_{\mathbf{R}_j})$. The basic idea behind this approach is that this pair would be the most wasteful in terms of the area covered by the MBR, if they are placed in the same node.

Given the seed, the next step is to determine the next rectangle/RC/MBR/MBRC to assign to one of the nodes. For each remaining rectangle/RC/MBR/MBRC the algorithm calculates the increase in area of the covering rectangle/RC/MBR/MBRC for the assignment nodes and furthermore takes the difference between them. Finally, the algorithm chooses the rectangle/RC/MBR/MBRC which has the maximum difference as it has a tendency to join one node rather than the other. This process is carried out till all the rectangles/RC have been assigned.

Deletion

Another important operation on the Rtree hierarchy is deletion of a specific entry. The removal of a specific entry in a node requires propagation of the changes throughout the Rtree hierarchy. The objective of this propagation is to resize the affected MBR/MBRC to tightly cover the rectangles/RC in the leaf nodes after the removal process. During this process if the number of entries in a node become less the minimum number of entries

d, that specific node is deleted and all the rectangles/RC are re-inserted into the Rtree hierarchy. If any node overflows during this re-insertion the node is split using the quadratic splitting algorithm as explained above.

2.4.2 Characteristics of the Rtree structure and Hierarchy Construction

This subsection explains the Rtree hierarchy construction process using a simple 2D example. As discussed in the previous section, the most important operation in the Rtree hierarchy construction is the rectangle insertion process. A rectangle is inserted into the Rtree structure through the process of *least expansion*. If the number of entries in a node exceeds M during the insertion process, the node has to be split. An exemplary Rtree structure is shown in Figure 2.3(b) assuming that each node can have a maximum of 2 entries ($M = 2$). The Rtree structure initially consists of a single node when rectangles A and B are inserted. Furthermore, if rectangles C and D are added the node splits increasing the height of the structure and forms overlapping MBR (E and F), as shown in Figure 2.3(b). The important aspect is that the MBR of the inner nodes in the Rtree structure can *overlap*. As a consequence of overlaps within the tree hierarchy multiple subtrees, i.e. child nodes, might need to be searched during a spatial query or search process. The maximum number of entries allowed per node M is another important factor in the hierarchy construction process. For a fixed number of leaf nodes increasing M generates tree structures containing fewer inner nodes but creates more overlaps. Consider the scenario shown in Figure 2.3 in which the assumption of 2 entries per node is considered. If the maximum number of entries per node is increased from 2 to 4, one node is required in the hierarchy to represent all the rectangles thereby reducing the number of nodes required for representation. The focus of this chapter is on 3D mapping, hence the term rectangular cuboids (RC) will be used for entries in the leaf nodes and MBRC for the entries within inner and root nodes.

2.5 Rtree Occupancy Grid

This section is divided into two subsections. The first subsection deals with the description of the occupancy grid based on the Rtree data structure whereas the second describes the adaptation of the grid cell resolution.

2.5.1 Occupancy Grid Formulation

The Rtree occupancy grid is probabilistic in nature and models the occupancy of its grid cells based on sensor observations. The proposed approach utilizes a specific sensor model to *incrementally* generate the grid structure and update the tree hierarchy composed of axis aligned rectangular cuboids. This chapter focuses on two different sensor models specifically the *beam-based* as well as the *beam-end point* model, whose characteristics are discussed in the following section. Depending on the sensor model, grid cells are either initialized and updated at the beam end points or also along the path followed by the beam. Figure 2.4 shows the grid cell initialization process for the *beam-based* sensor model in context of the Rtree based occupancy grid in comparison to the standard occupancy grid whose grid structure is predefined. Initially all grid cells i.e. entries in a leaf node of the Rtree occupancy grid are of cubic volume based on the chosen resolution of the grid, axis aligned and do not overlap. However, the inner nodes MBRC can overlap as discussed in the previous section.

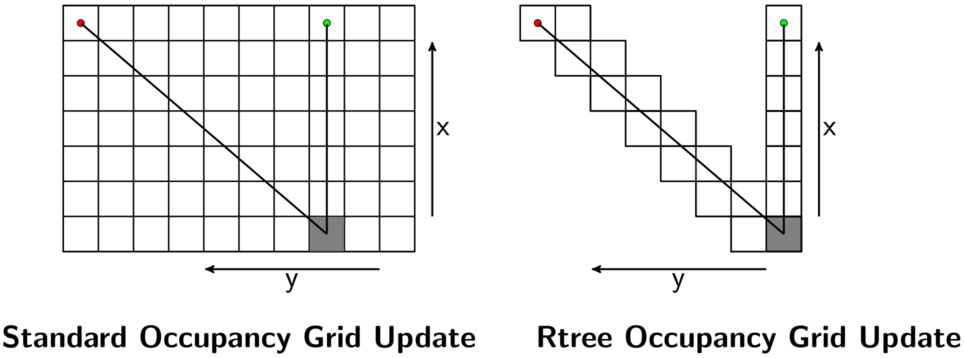


Fig. 2.4: The cell initialization process for the Rtree based Occupancy grid in context of the beam based sensor model. The standard occupancy grid has a predefined structure in which cubic grid cells are initialized with 0.5 occupancy probability in a fixed region. In contrast, the Rtree based occupancy grid incrementally generates the grid structure as sensor observations are obtained. The robot is shown as a solid grey block.

Let $\hat{z}_t$ represent the sensor observation, where the subscript denotes the time index. Consider a grid $G_t = \{g_1, g_2, \dots, g_n\}$ at time t consisting of n cubic or variable resolution grid cells g_i, $i = 1, \dots, n$. The notation used in this section is valid for cubic and variable resolution representations. Initially, the occupancy grid is composed of cubic grid cells, however as sensor observations are obtained the adaptation of grid cell resolution takes place (see Section 2.5.2) based on occupancy probabilities leading to variable resolution cells. The occupancy probability of any entry in the leaf node g_i representing the i^{th} grid cell can be derived from the posterior distribution over the cells given all the sensor observations $\hat{z}_{1:t}$ and robot poses $\boldsymbol{\zeta}_{1:t}$

$$P(g_1, g_2, \dots, g_n | \hat{z}_{1:t}, \boldsymbol{\zeta}_{1:t}).$$

A common assumption in the standard occupancy grid to reduce the dimensionality and computational complexity of the problem is to calculate the occupancy probability of a cell independently of other grid cells

$$P(g_1, g_2, \dots, g_n | \hat{z}_{1:t}, \boldsymbol{\zeta}_{1:t}) = \prod_{i=1}^{n} P(g_i | \hat{z}_{1:t}, \boldsymbol{\zeta}_{1:t}).$$

Furthermore, by transforming the observations based on the pose estimates into the global frame of reference we can omit the pose information

$$\prod_{i=1}^{n} P(g_i|\hat{z}_{1:t}, \boldsymbol{\zeta}_{1:t}) = \prod_{i=1}^{n} P(g_i|z_{1:t}),$$

where $z_{1:t}$ represents the transformed observations in the global frame of reference. By using the Bayes rule and incorporating the Markov assumption in the first term of the numerator, i.e. the current observation z_t is conditionally independent of previous observations $z_{1:t-1}$ given the robot pose, each grid cell probability can be written as

$$P(g_i|z_{1:t}) = \frac{P(z_t|g_i)P(g_i|z_{1:t-1})}{P(z_t|z_{1:t-1})}. \tag{2.1}$$

Similarly, the first term of the numerator in (2.1) can be written as

$$P(z_t|g_i) = \frac{P(g_i|z_t)P(z_t)}{P(g_i)}. \tag{2.2}$$

By substituting (2.2) into (2.1),

$$P(g_i|z_{1:t}) = \frac{P(g_i|z_t)P(z_t)P(g_i|z_{1:t-1})}{P(g_i)P(z_t|z_{1:t-1})}, \tag{2.3}$$

the equation defining the occupancy probability of cell g_i is derived. The probability that the cell g_i is free can then be similarly calculated as

$$1 - P(g_i|z_{1:t}) = \frac{(1 - P(g_i|z_t))P(z_t)(1 - P(g_i|z_{1:t-1}))}{(1 - P(g_i))P(z_t|z_{1:t-1})}. \tag{2.4}$$

Dividing (2.3) by (2.4) gives the odds,

$$\frac{P(g_i|z_{1:t})}{1 - P(g_i|z_{1:t})} = \frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1 - P(g_i))}{(1 - P(g_i|z_t))(1 - P(g_i|z_{1:t-1}))P(g_i)}, \tag{2.5}$$

which can be simplified by simple algebraic manipulation as

$$P(g_i|z_{1:t}) = (1 - P(g_i|z_{1:t}))\left[\frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1 - P(g_i))}{(1 - P(g_i|z_t))(1 - P(g_i|z_{1:t-1}))P(g_i)}\right],$$

$$\begin{aligned} P(g_i|z_{1:t})\left[1 + \frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1 - P(g_i))}{(1 - P(g_i|z_t))(1 - P(g_i|z_{1:t-1}))P(g_i)}\right] \\ = \frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1 - P(g_i))}{(1 - P(g_i|z_t))(1 - P(g_i|z_{1:t-1}))P(g_i)}, \end{aligned}$$

$$P(g_i|z_{1:t}) \left[\frac{(1-P(g_i|z_t))(1-P(g_i|z_{1:t-1}))P(g_i) + P(g_i|z_t)P(g_i|z_{1:t-1})(1-P(g_i))}{(1-P(g_i|z_t))(1-P(g_i|z_{1:t-1}))P(g_i)}\right]$$
$$= \frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1-P(g_i))}{(1-P(g_i|z_t))(1-P(g_i|z_{1:t-1}))P(g_i)}.$$

Shifting the left hand side terms to the right gives

$$P(g_i|z_{1:t}) = \frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1-P(g_i))}{(1-P(g_i|z_t))(1-P(g_i|z_{1:t-1}))P(g_i) + P(g_i|z_t)P(g_i|z_{1:t-1})(1-P(g_i))},$$

and inverting the right hand side gives

$$P(g_i|z_{1:t}) = \left[\frac{P(g_i|z_t)P(g_i|z_{1:t-1})(1-P(g_i)) + (1-P(g_i|z_t))(1-P(g_i|z_{1:t-1}))P(g_i)}{P(g_i|z_t)P(g_i|z_{1:t-1})(1-P(g_i))}\right]^{-1},$$

that can be easily transformed into [73, 125, 200]

$$P(g_i|z_{1:t}) = \left[1 + \frac{1-P(g_i|z_t)}{P(g_i|z_t)} \frac{1-P(g_i|z_{1:t-1})}{P(g_i|z_{1:t-1})} \frac{P(g_i)}{1-P(g_i)}\right]^{-1}, \tag{2.6}$$

which is a commonly used inverse sensor model in robotic mapping. $P(g_i|z_{1:t})$ represents the occupancy probability of the i^{th} grid cell given all observations. $P(g_i)$ represents the occupancy probability of a grid cell prior to any observations. $P(g_i|z_t)$ and $P(g_i|z_{1:t-1})$ represent the probability given the most current observation z_t and observations since the beginning of time until time $t-1$ respectively. Eq (2.5) can be also be converted to log odds form to simplify the computation as it reduces the occupancy update to a simple addition operation

$$l(g_i|z_{1:t}) = l(g_i|z_t) + l(g_i|z_{1:t-1}) - l_o, \tag{2.7}$$

where $l(g_i) = \log\left[\dfrac{P(g_i)}{1-P(g_i)}\right]$ is the log-odds form whereas l_o represents the prior which is the same for every cell. In literature [73, 200], occupancy grids use a probability clamping threshold to prevent each cell of being over confident about its state. Following the same pattern, the Rtree based occupancy grid defines a minimum and maximum probability (or log-odds) threshold $\mu_{\min}$, $\mu_{\max}$ respectively after which a grid cell is no longer updated. The following subsection provides details about the inverse sensor models considered in this chapter as well as their properties.

Inverse Sensor Models

The proposed Rtree based occupancy grid is a generic approach to model a grid, hence it can be used with a wide range of inverse sensor models which define different update rules. The term $P(g_i|z_t)$ in (2.6) or $l(g_i|z_t)$ in (2.7) determine how these updates are carried out. In the domain of the robotics, the following inverse sensor models are commonly used

- Beam-based model
- Beam-end point model

Beam-based Model:
The beam-based approach models the physical properties of a beam, hence it considers the complete path traversed by the beam along with the beam end point which corresponds to an object/surface detected by the sensor. To be more specific a beam based model traces a ray through the grid [2, 18] and updates the end point as being occupied and the path of the beam as free space. Mathematically the above mentioned update is written as

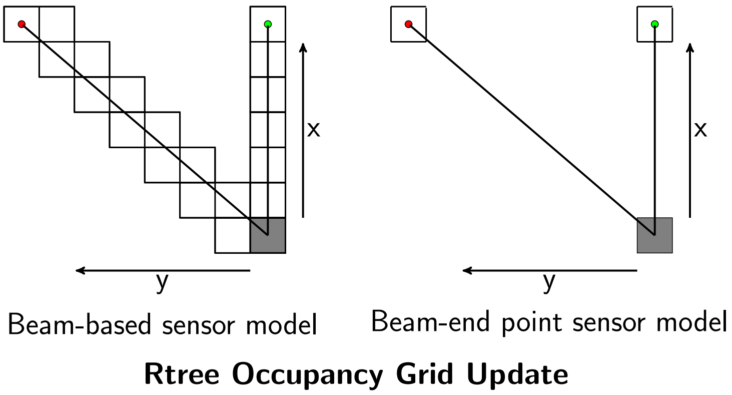
$$P(g_i|z_t) = \begin{cases} P_{\text{occ}} & \text{if beam is reflected within volume} \\ \\ P_{\text{free}} & \text{if beam traversed volume} \end{cases},$$

where the terms P_{occ} and P_{free} are dependent on the sensor properties. Figure 2.4 shows an example of the grid cell update process for a beam-based sensor model.

Beam-end point Model:
As beam-based models tend to consider the complete path of the beam, they can be computationally expensive. To reduce computational cost, an alternative approach can be to ignore the path of the beam. Hence, beam-end point models tend to update the end points of the beam while ignoring the complete path traversed by the beam. Mathematically this update is written as

$$P(g_i|z_t) = P_{\text{occ}} \quad \text{if beam is reflected within volume.}$$

Figure 2.5 shows the difference in update between the beam based and the beam-end point sensor model.



Beam-based sensor model

Beam-end point sensor model

Rtree Occupancy Grid Update

Fig. 2.5: Comparison of the beam-based and the beam-end point sensor model. The robot position is depicted as a solid square.

2.5.2 Resolution Adaptation Process

The previous section focused on the creation, update process of the Rtree occupancy grid whereas this section describes the incremental resolution adaptation within the grid using the fusion process based on occupancy probabilities. The adaptation of the cell resolution is the process of reducing the number of cells required to represent the environment. Given a grid $G_t = \{g_1, g_2, \ldots, g_n\}$ at time index t consisting of n cubic or variable resolutions grid cells, the objective of the resolution adaptation process is to generate a grid $G_{t+1} = \{\bar{g}_1, \bar{g}_2, \ldots, \bar{g}_m\}$ (the bar indicating a modification of cell size) where $m \ll n$ by allowing the cells to fuse. This section focuses on the resolution adaptation process of the cells given the sensor models defined in the previous section i.e. beam-based or beam-end point based sensor model.

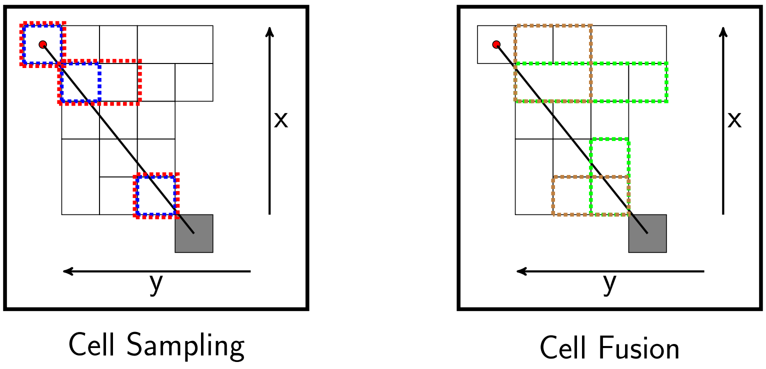


Fig. 2.6: (Best viewed in color) The cell sampling and fusion process for the beam-based sensor model. In context of the beam-end point sensor model, only the end points are allowed to expand and fuse. (a) The process of sampling cells along the beam path to allow fusion in the occupancy grid. The randomly sampled cells are shown with blue dashes and the corresponding cells of the grid are shown with a pattern of red dashes. (b) The cell expansion process for two cases i.e. cube and a rectangular cuboid, shown in (a). The search direction is defined by the cell width vector $\mathbf{w}_i$. The first preference is shown with green dashes followed by the second preference shown in beige. If all sides of the cell are the same i.e. the cubic cell closest to the robot, a fixed search direction is employed (first along the x axis and then along the y axis). In case of the rectangular cuboid, the expansion is biased given the larger side of the cuboid, as shown in the figure.

Beam-based Sensor Model

The grid cell resolution adaptation process consists of two basic steps, firstly selection of the cells that are allowed to expand and furthermore the expansion, fusion process with neighborhood cells. The cell selection, expansion and fusion process are explained in detail in the following subsections.

Cell Selection

An important aspect within the fusion process is the selection of the cell g_i which is allowed to expand and fuse with its neighbourhood cells. In context of beam-based sensor model it is possible to allow all grid cells at the beam end point and along the beam path (given the sensor observations) to fuse. However, this strategy causes a substantial increase in the computational cost, thus a different strategy is adopted. Consider the sensor observation $z_t = \{z_t^1, z_t^2, \ldots, z_t^n\}$ where z_t^i represents the i^{th} observation among the n point observations from a laser scanner at time index t. The occupancy grid updates

```
Fuse($g_i$)
Input: $g_i$ // cell $g_i$ to be expanded
Outcome: $\bar{g}_i = (\bar{\mathbf{R}}_i \; P(\bar{g}_i|z_{1:t}))$ or
            fusion not possible
Procedure:
Determine the width vector $\mathbf{w}_i$ of $g_i$;
If (all elements of $\mathbf{w}_i$ of $g_i$ are equal)
   $\bar{\mathbf{S}}_i = \{\bar{\mathbf{S}}_i^x, \bar{\mathbf{S}}_i^y, \bar{\mathbf{S}}_i^z, \bar{\mathbf{S}}_i^{-x}, \bar{\mathbf{S}}_i^{-y}, \bar{\mathbf{S}}_i^{-z}\}$;
   //first expand along x, then y etc.
else
   Re-arrange $\bar{\mathbf{S}}_i$ based on $\mathbf{w}_i$;
for-all $j$ ($j \leq |\bar{\mathbf{S}}_i|$) // $|\bar{\mathbf{S}}_i|$ is the number
                                   of elements in $\bar{\mathbf{S}}_i$
   $\bar{\mathbf{R}}_i = \mathbf{R}_i +^j \bar{\mathbf{S}}_i$;
   $\forall k$ such that $\mathbf{R}_k$ is contained in $\bar{\mathbf{R}}_i$
      If $((P(g_i|z_{1:t})$ and $P(g_k|z_{1:t})) \leq \mu_{min}$
      or $(P(g_i|z_{1:t})$ and $P(g_k|z_{1:t})) \geq \mu_{max}$ )
         Fuse cells to form $\bar{g}_i$;
         Remove $g_i$ and $g_k$ from the grid;
         Fuse($\bar{g}_i$); // recursive call
         return;
 end for;
 return;
```

Fig. 2.7: The pseudocode describing the fusion process of the grid cells of the occupancy grid

the cell corresponding to the beam end point of the observation z_t^i and all cells that lie along the beam path. Given all beam end point observations (z_t^i, $i = 1, \ldots, n$) a set $T = \{g_1, \ldots, g_p\}$ composed of grid cells can be generated by randomly sampling cells along the beam path based on the beam length and always considering the beam end point. Figure 2.6 (left image) shows an illustration of the cell selection and the generation process of the set T.

Cell Expansion and Fusion:

The grid cells within the set T obtained from the cell selection process are allowed to expand and fuse with the neighbourhood cells. The pseudocode of the expansion and fusion process for any grid cell g_i in the set T is shown in Figure 2.7 and explained in detail here. The fusion process shown in Figure 2.7 is carried out after every sensor observation. As mentioned in Section 2.4 each grid cell g_i (or each entry in the leaf node) contains the following

$$g_i = \left(\mathbf{R}_i \quad P(g_i|z_{1:t})\right),$$

where $\mathbf{R}_i = \left[\mathbf{r}_i^{\min} \quad \mathbf{r}_i^{\max}\right]^T$ and $P(g_i)$ represents the occupancy probability. In the context of Rtree based occupancy grid, $\mathbf{r}_i^{\min} = \left[x_i^{\min} \quad y_i^{\min} \quad z_i^{\min}\right]$ and $\mathbf{r}_i^{\max} = \left[x_i^{\max} \quad y_i^{\max} \quad z_i^{\max}\right]$

represents the minimum and maximum bounds of the axis aligned rectangular cuboid in the global frame of reference. Given $\mathbf{r}_i^{\min}$ and $\mathbf{r}_i^{\max}$, the width vector $\mathbf{w}_i = \begin{bmatrix} w_i^x & w_i^y & w_i^z \end{bmatrix}$ can be easily extracted. The expansion process of the cell g_i in the Rtree based occupancy grid is defined (line 8 of Figure 2.7) as

$$\bar{\mathbf{R}}_i = \mathbf{R}_i +^j \bar{\mathbf{S}}_i,$$

for any specific search direction index j, where $\bar{\mathbf{S}}_i$ represents the search direction set. To explain the notation consider that $\bar{\mathbf{S}}_i = \{\bar{\mathbf{S}}_i^x, \bar{\mathbf{S}}_i^y, \bar{\mathbf{S}}_i^z, \bar{\mathbf{S}}_i^{-x}, \bar{\mathbf{S}}_i^{-y}, \bar{\mathbf{S}}_i^{-z}\}$, which states that the i^{th} grid cell should try to expand along the x axis, then along the y axis etc. The index j in ${}^j\bar{\mathbf{S}}_i$ represents the j^{th} element of the set $\bar{\mathbf{S}}_i$, hence the index $j = 1$ would correspond to $\mathbf{S}_i^x$ in the above example. The search direction set for any specific cell g_i is chosen based on the width vector $\mathbf{w}_i$. If all sides of the axis aligned rectangular cuboid are equal, a fixed set of search directions (line 2-3 of Figure 2.7) is chosen otherwise it is biased based on the larger side of the rectangular cuboid (line 6 of Figure 2.7), as shown in Figure 2.6. The exact form of the search direction $\bar{\mathbf{S}}_i^x$ is defined below

$$\bar{\mathbf{S}}_i^x = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{W}_i^x \end{bmatrix} \begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{\sigma}_i^x \end{bmatrix} \tag{2.8}$$

where $\mathbf{0}_{m\times n}$ represents a zero matrix of m rows, n columns and $\mathbf{W}_i^x$ is a 3×3 matrix defined as

$$\mathbf{W}_i^x = \begin{bmatrix} \mathbf{w}_i \\ \mathbf{0}_{2\times3} \end{bmatrix},$$

and $\boldsymbol{\sigma}_i^x$ is a 3×1 unit vector (scaled based on width of rectangular cuboid) along the x dimension of the global reference frame. The basic operation being performed in (2.8) is the modification of the maximal x bound of the axis aligned rectangular cuboid. In a similar manner other search directions such as $\bar{\mathbf{S}}_i^y$, $\bar{\mathbf{S}}_i^z$, $\bar{\mathbf{S}}_i^{-x}$ etc. can be defined by replacing $\mathbf{W}_i^x$, $\boldsymbol{\sigma}_i^x$ and manipulating the structure of matrices (to change the maximum or minimum bound of the rectangular cuboid). Given the expanded cell $\bar{\mathbf{R}}_i$ (line 8 of Figure 2.7) based on the search direction, fusion with *neighbouring cells* g_k is allowed if (line 9 of Figure 2.7)

$$\forall k \text{ such that } \mathbf{R}_k \text{ is contained in } \bar{\mathbf{R}}_i, \tag{2.9}$$

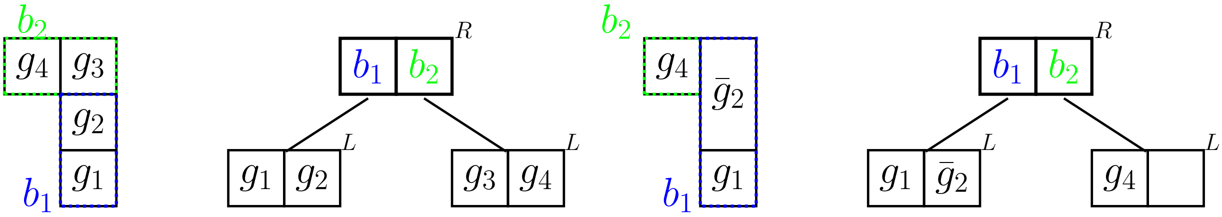
any of the following two conditions is satisfied (line 10 of Figure 2.7)

$$\forall k : P(g_k|z_{1:t}) \leq \mu_{\min} \text{ and } P(g_i|z_{1:t}) \leq \mu_{\min}, \tag{2.10}$$

or

$$\forall k : P(g_k|z_{1:t}) \geq \mu_{\max} \text{ and } P(g_i|z_{1:t}) \geq \mu_{\max}. \tag{2.11}$$

Equation (2.9) simply states that all rectangular cuboids $\mathbf{R}_k$ should be contained in the expanded rectangular cuboid $\bar{\mathbf{R}}_i$, whereas (2.10) and (2.11) state that the occupancy probability of each cell g_k should be below $\mu_{\min}$ or above $\mu_{\max}$ if the occupancy probability of cell g_i is below $\mu_{\min}$ or above $\mu_{\max}$ respectively. The objective of the constraints (2.10) and (2.11) is to limit the fusion to only those cells which have a high probability of being



(a) Initial state of the Rtree based occupancy grid

(b) Final state after the fusion process

Fig. 2.8: An example scenario depicting the hierarchy adaptation of the Rtree occupancy grid based on the fusion process (colors have been added to aid visualization of the tree hierarchy). g_i represents the grid cells in the hierarchy and b_1, b_2 represent the MBR. The following assumptions are made for the example scenario shown above: Firstly the probability of only cell g_2 and g_3 is above $\mu_{\max}$ and the cell g_2 is chosen for expansion and tries to expand in the direction of cell g_3. Secondly, the value of M is assumed to be 2 (as in Figure 2.3) a) The initial state of the hierarchy of the Rtree occupancy grid. b) The final state of the hierarchy after removal of expanding cell g_i ($i = 2$), g_k ($k = 3$) and insertion of the fused grid cell $\bar{g}_2$ (where $\mathbf{R}_2$, $\mathbf{R}_3$ is contained in $\bar{\mathbf{R}}_2$).

occupied or free and are no longer being updated as they are beyond the clamping thresholds $(\mu_{\min}, \mu_{\max})$. If the conditions stated above are satisfied the cells are fused to form $\bar{g}_i = (\bar{\mathbf{R}}_i\ P(\bar{g}_i|z_{1:t}))$. Additionally, cell g_i and cells g_k ($\forall k$ such that $\mathbf{R}_k$ is contained in $\bar{\mathbf{R}}_i$) are removed from the grid. The probability of the fused cell is taken as an average probability of cells g_k ($\forall k$ such that $\mathbf{R}_k$ is contained in $\bar{\mathbf{R}}_i$) that are contained in it (all occupancy probabilities are above $\mu_{\max}$ or below $\mu_{\min}$ based on (2.10) or (2.11)). The fusion function is called recursively (line 13 of Figure 2.7) after merging the cells to form $\bar{g}_i$. In case fusion is not possible (line 15 of Figure 2.7), the algorithm returns without any modification in the cell size. This fusion process continues for all the elements of the set T. The description mentioned above focuses on the fusion of grid cells i.e. leaf nodes of the Rtree occupancy grid, however, the incremental adaptation process also causes a change in the tree hierarchy after every successful fusion. Figure 2.8 shows an example scenario depicting the hierarchy adaptation due to the incremental fusion process. Once a specific number of neighbouring cells g_k ($\forall k$ such that $\mathbf{R}_k$ is contained in $\bar{\mathbf{R}}_i$) along with the expanding cell g_i have been chosen for fusion, they are first removed from the tree hierarchy (line 12 of Figure 2.7) which causes a change in the size of MBRC being propagated up the hierarchy till the root. Additionally, a node might underflow (the number of entries might fall below d, see Section 2.4) during this removal process; hence that specific node is removed and all entries in that node are reinserted into the hierarchy based on the least expansion principle (see Section 2.4). After the cells have been removed, the fused grid cell $\bar{g}_i$ is also inserted into the hierarchy based on the least expansion principle.

Beam-end point Sensor Model

The following subsections provide details in the modification of the cell expansion and fusion process for beam-end point model in comparison to the beam-based sensor model.

Cell Selection:
The grid cell corresponding to the beam-end point is always selected and allowed to expand, fuse with neighboring cells. Hence given the i^{th} sensor observation at time t, i.e. z_t^i, the corresponding cell g_i is selected and the set T is only composed of the beam end point observation.

Cell Expansion and Fusion:
Given the selected cells, the cell expansion and fusion process is a simplified version of the pseudocode in Figure 2.7 in which the condition on line 10 is modified to

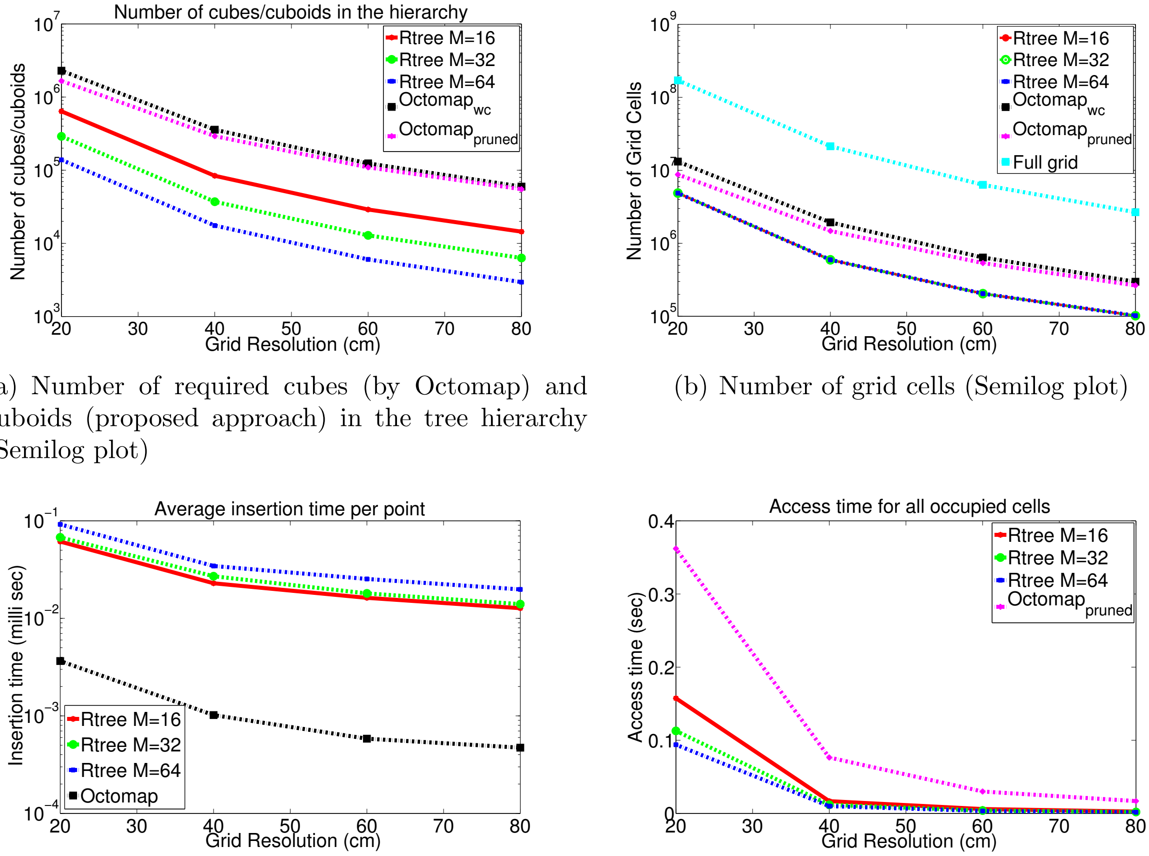
$$\textbf{if } P(g_i|z_{1:t}) \textbf{ and } P(g_k|z_{1:t}) \geq \mu_{max}$$

and the condition for the free cells stated in (2.10) is ignored. The hierarchy adaptation and the cell insertion and deletion process is carried out in the same manner as shown graphically in Figure 2.8 and discussed in the beam-based sensor model.

2.5.3 Recursion

After the adaptation process, the new sensor observation z_{t+1} is used to update the occupancy values of the current grid G_{t+1} (Section 2.5.1) followed by another resolution adaptation or fusion step (Section 2.5.2). This recursive formulation of occupancy update and cell fusion continues for all sensor observations obtained by the robot. The search direction strategy shown in Figure 2.7 is chosen as it leads to the best results on the Freiburg campus dataset[1] which is used for evaluation in the experimental section. In principle the success of a specific search strategy for fusion is dependent on the structure of the environment which is unknown prior to the mapping process. In the worst case scenario if no fusion takes place in occupied and free regions (which is highly unlikely as free space does not have any definite shape and can be fused) the number of grid cells required by the Rtree based adaptive occupancy grid and the standard occupancy grid are the same (as the Rtree based adaptive occupancy grid initially contains cubic grid cells). The incremental fusion process presented in this paper is not restricted to any specific search strategy, rather it can be changed as per requirements or based on any prior information available about the environment. In the proposed approach $\mu_{\min}$ and $\mu_{\max}$ are set to very low and high occupancy probabilities respectively to ensure that stable regions of the occupancy grid are fused given a static world assumption. In addition, the fused regions of the Rtree based adaptive occupancy grid are constrained to be axis aligned. However, this axis aligned constraint is inherent to virtually all grid based representations as they are composed of axis aligned cubes.

[1] Courtesy of B. Steder and R Kümmerle, available at http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/



(a) Number of required cubes (by Octomap) and cuboids (proposed approach) in the tree hierarchy (Semilog plot)

(b) Number of grid cells (Semilog plot)

(c) Average insertion time per point (Semilog plot)

(d) Access time for occupied cells only, given the complete tree hierarchy consisting of occupied and free cells

Fig. 2.9: Results on all 81 scans of the Freiburg campus dataset using the beam-based sensor model. (a) The number of cubes and cuboids required by Octomap and the Rtree based adaptive occupancy grid in the tree hierarchy (not including the leaves nodes). An increase in parameter M causes the Rtree to effectively reduce the number of cuboids required for representation in the hierarchy. (b) The number of grid cells as a function of the grid resolution shown as a semilog plot. The grid cells (leaf nodes) required by the Rtree based adaptive occupancy grid is less than Octomap due to the incremental fusion process which leads to axis aligned rectangular cuboids. (c) The average insertion time (per point) of the Octomap and the Rtree based adaptive occupancy grid. (d) Access times for occupied grid cells only given the entire hierarchy consisting of free and occupied grid cells.

2.6 Experimental Evaluation

This section presents an experimental evaluation of the proposed approach for different inverse sensor models discussed in section 2.5.1. In addition, this section also highlights the influence of the maximum number of entries per node, i.e. M, on the Rtree hierarchy construction process. The proposed adaptive occupancy grid is compared to the Octomap [73, 200] (version 1.6.1) approach on the Freiburg campus dataset. The important aspects such as occupancy thresholds i.e. $\mu_{\text{min}} = 0.12$, $\mu_{\text{max}} = 0.97$ and inverse sensor models parameters i.e. $P_{\text{occ}} = 0.7$, $P_{\text{free}} = 0.4$ are the same as mentioned in [73] and were fixed for all experiments discussed in this section. The evaluation is based on the insertion, access time as well as the number of grid cells required for 3D representation. The insertion time is defined as the time required to insert all laser scans into the grid. In context of the Rtree based adaptive occupancy grid it also includes the time taken by the fusion process of the grid cells. The access time of Octomap and the Rtree based adaptive occupancy grid corresponds to the time taken to access only the occupied grid cells given the entire hierarchy after all scans have been inserted. The *graph2tree* tool (provided along with Octomap implementation) is used to determine the number of inner, leaf nodes and insertion time of Octomap. The access time of Octomap is determined by using the iterator based access method (on the pruned Octomap) after all scans were inserted. The evaluation is performed on a single core of an Intel i5-2500K, 3.3 GHz processor with 16 GB RAM.

2.6.1 Beam based Model

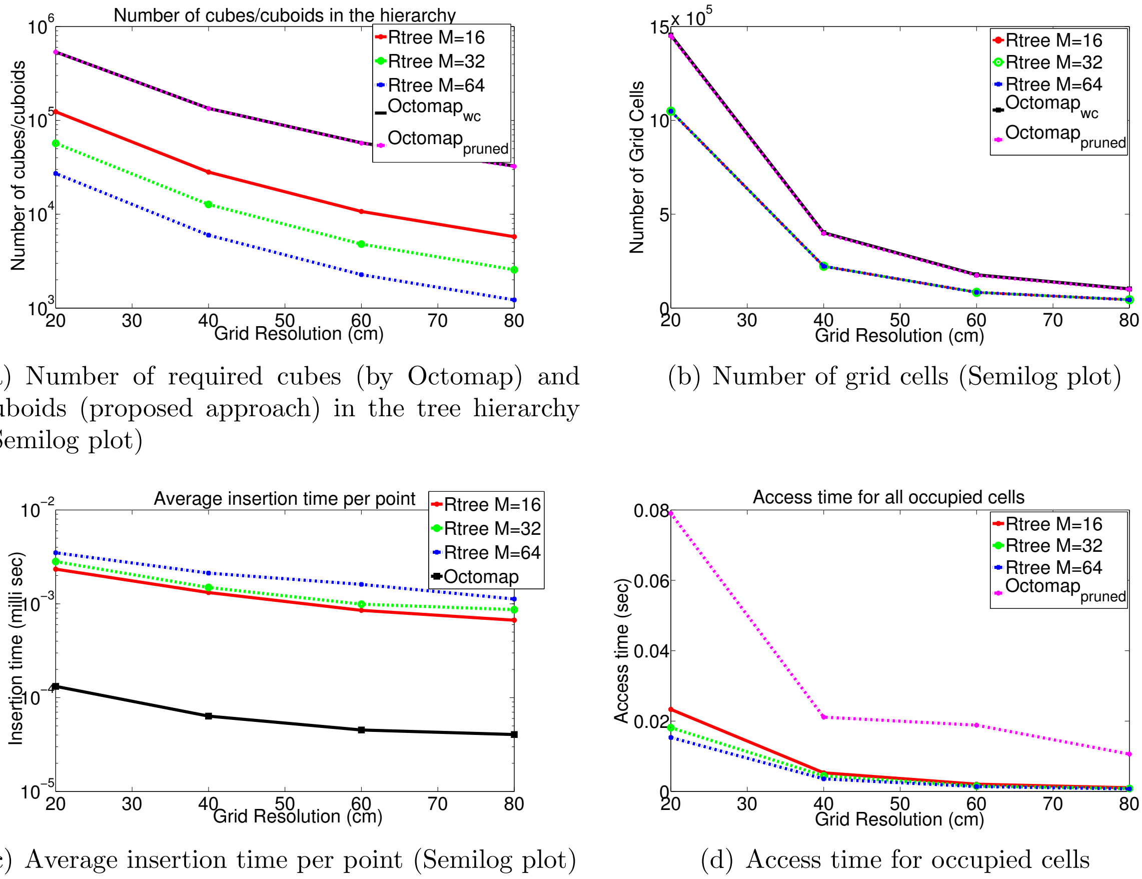
Figure 2.9 shows the results of the evaluation on all 81 scans of the Freiburg campus dataset based on the insertion, access time as well as the number of grid cells required by both approaches. Figure 2.9(a) shows the number of cuboids required in the tree hierarchy by the Rtree based occupancy grid in comparison to the cubes required by Octomap. In case of the Rtree occupancy grid, the evaluation is shown for different branching factors M whereas the Octomap evaluation is shown for two cases i.e. *without compression (wc)* and *pruned*. In contrast, Figure 2.9(b) shows the number grid cells required by both approaches. It can be seen through the above mentioned figures that the Rtree based adaptive occupancy grid requires fewer grid cells as well as cuboids in the tree hierarchy in comparison to the cubes and grid cells of the Octomap approach. Focusing on cubes/cuboids required in the tree hierarchy first, two main reasons can be attributed to this, firstly Octomap (based on Octrees) has a pre-defined hierarchy consisting of cubes with the number of children per node fixed to 8. In contrast, the nodes in the Rtree based occupancy grid can contain an arbitrary maximum number of children (M) which can effectively reduce the number of nodes required (as discussed in Section 2.4). Secondly, the MBRC in the Rtree based adaptive occupancy grid hierarchy can overlap and are not constrained to be cubic. Considering the number of grid cells required for representation as shown in Figure 2.9(b), the comparison between pruned Octomap and the Rtree based adaptive occupancy grid is interesting. The Octomap approach uses the μ_{min} and μ_{max} threshold to prune out regions of the Octree hierarchy (nodes and leaves) to achieve compression whereas the Rtree based adaptive occupancy grid approach uses these parameters for fusion of entries in the leaf nodes only. The reduction in the number of grid cells required to represent the environment by the Rtree based adaptive occupancy grid in contrast to pruned Octomap is **28.51**% at a 20 cm resolution grid. The amount of grid cells required by the full 3D grid, or standard occupancy grid, as shown in Figure 2.9(b) is calculated based on [73] $\frac{x \times y \times z}{r^3}$, where x, y and z is the minimal bounding box in each dimension (292 × 167 × 28 m for the Freiburg campus dataset) and r represents the resolution of the grid in meters. It is important to specify here that for a fixed grid resolution the maximum number of entries per node M

does not influence the number of grid cells required for representing the environment nor the fusion process. A comparison with the maximum likelihood compression of Octomap is not performed in this chapter as it involves thresholding (either occupied or free) all the nodes of the Octree. Due to its lossy nature, this thresholding process might lead to an inaccurate environment representation. In addition, the occupancy probability is essential for the Rtree based adaptive occupancy grid as the entire resolution adaptation process is based on it. Consequently, this would prevent probabilistic fusion of grid cells in case the robot receives additional sensor observations.

Figure 2.9(c) shows the normalized insertion time per point of the Rtree based adaptive occupancy grid and Octomap. The Rtree based adaptive occupancy grid is slower than Octomap due to multiple reasons. The Rtree based adaptive occupancy grid incrementally generates the tree hierarchy based on node splitting and least expansion as observations are obtained whereas Octomap has a predefined hierarchy consisting of cubes. Additionally, due to the fusion process the tree hierarchy of the Rtree based adaptive occupancy grid needs to be regularly updated. The variation in grid cells (entries in the leaf nodes) is propagated up the hierarchy leading to a change in the MBRC of the inner branches. Finally, the overlaps between the MBRC of the inner branches in the Rtree based occupancy grid can also slow down the query/search process. An increase in parameter M increases the tree width as well as the insertion time because of increased overlaps between MBRC. Figure 2.9(d) shows the time required to access all the occupied cells in the grid given the entire hierarchy composed of occupied and free grid cells. It can be seen that the Rtree based adaptive occupancy grid is capable of accessing the occupied cells faster than Octomap. An increase in parameter M also reduces the number of nodes required in the Rtree hierarchy and causes the access time of occupied cells to decrease as can be seen in Figure 2.9(d). Figure 2.11 shows examples of the axis aligned rectangular cuboids generated by the Rtree based adaptive occupancy grid for the occupied regions on the Freiburg campus dataset. The fused free space regions are not shown in the figure for the ease of visualization.

2.6.2 Beam end-point Model

This subsection presents the results of the evaluation of the Rtree occupancy grid and the Octomap approach using the beam-end point sensor model discussed in Section 2.5.1. Figure 2.10 shows the results for the Rtree based occupancy grid and the Octomap approach using the same evaluation metrics used in the beam-based sensor model. It can be seen that the overall trend and conclusion are the same for the beam-based and the beam-end point sensor model. Figure 2.10(a) shows that the Rtree based occupancy grid requires less number of inner nodes in the hierarchy to represent the grid in comparison to the Octomap approach. In addition by comparing Figure 2.10(a) and Figure 2.9(a), it can be seen that the magnitude of inner nodes required for the beam-end point based model is less than the beam-based sensor model. The reason for this being that the beam-end point based model does not model free space. Figure 2.10(b) shows the number of grid cells required by both approaches. It can be seen that due to the cubic grid cell assumption the Octomap compression, i.e. pruned vs without compression, does not work well for the beam-end point model. This essentially highlights that the compression results visible in



(a) Number of required cubes (by Octomap) and cuboids (proposed approach) in the tree hierarchy (Semilog plot)

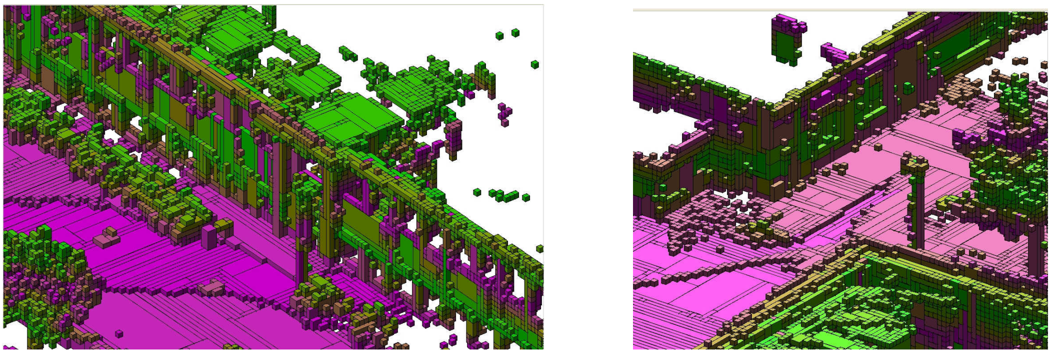
(b) Number of grid cells (Semilog plot)

(c) Average insertion time per point (Semilog plot)

(d) Access time for occupied cells

Fig. 2.10: Results on the Freiburg campus dataset using the beam-end point sensor model. (a) The number of cubes and cuboids required by Octomap and the Rtree based adaptive occupancy grid in the tree hierarchy (not including the leaves nodes). (b) The number of grid cells as a function of the grid resolution shown as a semilog plot. (c) The average insertion time (per point) of the Octomap and the Rtree based adaptive occupancy grid. (d) Access times for occupied grid cells.

Figure 2.9(b) for Octomap are mainly due to fusion in free space regions as it does not have a definite shape. It can also be seen that the number of grid cells required by the Rtree based occupancy grid is less than the Octomap approach as occupied regions can be effectively approximated by rectangular cuboids. In addition by comparing Figure 2.10(b) with Figure 2.9(b), a huge reduction is observed in the magnitude of grid cells required to represent the environment as the beam-end approach only models the occupied regions. As stated in the previous subsection and visible in the results of this section in Figure 2.10(c), the insertion time for the Rtree occupancy grid is worse than the Octomap approach due to the incremental construction, update of the hierarchy as well as the presence of overlaps between cuboids of inner nodes within the hierarchy. Similar to the beam-based sensor model, it can be seen in Figure 2.10(d), that the access times of the Rtree based occupancy grid are better than the Octomap approach.



(a) Visualization of fused occupied grid cells

(b) Visualization of fused occupied grid cells

Fig. 2.11: Visualization of fused occupied grid cells on the Freiburg campus dataset (colors have been assigned for the ease of visualization).

2.7 Conclusion and Future Work

This chapter proposes an approach which is capable of modeling the environment using a variable resolution grid. The variable resolution grid is stored in a hierarchy of axis-aligned rectangular cuboids, which is generated incrementally and adapted based on sensor observations. In addition, the presented approach is quite flexible as it allows the user to define the maximum number of entries per node thereby influencing its performance in terms of the number of nodes required in the hierarchy for representation as well as the insertion and access times. An extensive evaluation is carried out of the proposed approach in comparison to the state-of-the-art Octomap approach on a publicly available dataset. The evaluation shows that the proposed approach requires less number of grid cells to approximate the environment and furthermore allows faster access times of the occupied regions in the grid.

Future work includes an evaluation of the proposed approach in modeling dynamic environments. The scope of this thesis has been limited to static environments, however the proposed fusion process is easily extendable to environments containing dynamics. This extension is possible by splitting the fused cells of the dynamic region based on the chosen resolution of the grid, if the occupancy probability goes above or below the clamping threshold. Additionally, future work also includes an evaluation of different search strategies for the grid cell fusion process of the Rtree based adaptive occupancy grid.

3 Laser Intensities for SLAM

Summary and Contribution: *This chapter contributes in the domain of SLAM by proposing an approach that is capable of acquiring surface reflectivity characteristics from laser scanner observations for robot pose estimation and mapping. Hence this chapter discusses a simple calibration approach to acquire a pose-invariant measure of surface reflectivity from laser scanner observations. Furthermore, this reflectivity measure is embedded in an extension of the Hector SLAM algorithm which utilizes this information for pose estimation as well as acquiring a reflectivity map of the environment i.e. occupancy grid map augmented with surface reflectivity characteristics. An extensive experimental evaluation is carried out to highlight the advantages as well as attributes of the calibration approach and the proposed extension of the Hector SLAM algorithm.*

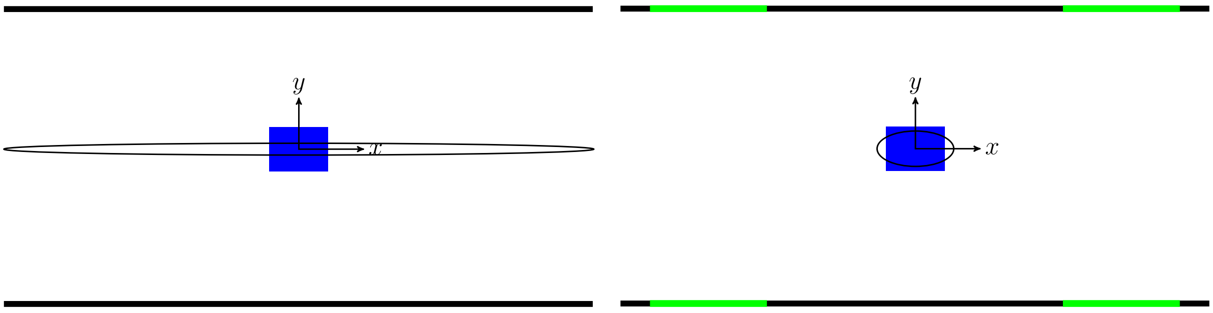
3.1 Introduction

The research work in the field of Simultaneous Localization and Mapping (SLAM) [59, 82, 90] has provided robots the capability of simultaneously estimating their own pose and acquiring an accurate topological/metric map of the environment. The previous chapter of this thesis focused on the aspect of environment representation, which provides the foundation for creating a map by defining the geometric primitive used for approximating the environment. In contrast this chapter focuses on SLAM, which couples the geometric primitive used for environment representation with the robot pose estimation process to allow online, incremental map generation of the environment based on sensor observations. An accurate map of the environment is essential requirement for a variety of robotic tasks such as global localization, navigation and exploration. SLAM has been an active research area in the field of robotics due to its application in the domain of autonomous driving, personal assistive robots etc. The SLAM algorithm consists of two core components: firstly the pose estimation and secondly the map creation process. A good pose estimate is required to generate an accurate map and at the same time an accurate map is essential for accurate pose estimation, hence SLAM is typically titled the *chicken-and-egg* problem.

The initial research focus of the robotics community within the domain of SLAM was on the development of filtering algorithms such as the extended Kalman filter (EKF) [39, 171]. The research community has focused on different aspects of EKF SLAM i.e. computational complexity [39, 151] as well as the consistency of the algorithm [4, 80]. The complexity of the EKF is $O(n^2)$ due to the covariance matrix update where n is the number of landmarks in the map. The EKF has been successfully applied for small scale environments, however the quadratic complexity limits its usage for environments containing a large number of features. To deal with this complexity different approaches have been proposed that rely on map update in local regions [62, 180]. Recently, the extended information filter [185, 194] has been proposed which takes advantage of the sparseness of the information matrix (i.e. inverse of the covariance matrix) to deal with the above mentioned issue. In addition, a divide and conquer mechanism [151] been proposed that has linear complexity in the number of landmarks in the map. The research community has also focused on the aspect of inaccuracies caused by the linearization of the EKF leading to the usage of the unscented Kalman filter (UKF) [112]. Another aspect of intense focus within the SLAM community has been to relax the Gaussian assumption associated with Kalman filters, as the mobile robot kinematics are nonlinear leading to non Gaussian distributions. To resolve this issue, particle filter [59, 122, 123] based approaches have been proposed that try to explicitly model the distribution using samples. The interest in the application of particle filters for SLAM has mainly been driven by the increase in computational power in the last few decades.

In contrast to the usage of *filtering* algorithms i.e. EKF, UKF or the particle filters, recently the research work in the robotics community has focused on the usage of smoothing algorithms for SLAM [36, 81, 82]. Along similar lines, different graph optimization based SLAM approaches have also been proposed [95, 186]. The majority of this work is inspired by the research on (sparse) bundle adjustment in the domain of computer vision and photogrammetry [66, 93, 105, 192]. In graph SLAM literature, the entire framework is typically divided into two components: the front-end and the back-end. The front-end deals with the raw sensor data and generates the graph structure by defining the node positions as well as edge constraints between nodes. These edge constraints can define two different cases: firstly the motion between consecutive robot poses and secondly the case when the robot returns to a previously visited location (loop closure constraints). The back-end takes these constraints and estimates the posterior distribution over the robot poses. In addition in context of SLAM, there also exist scan matching based approaches [37, 119, 143, 145], which can be sufficiently accurate for small scale mapping. Typical examples of such scan matching algorithms include iterative closest point (ICP) [157, 166], Normal distribution transform (NDT) [10, 109] as well as Hector SLAM [90]. These approaches typically estimate the transformation between consecutive robot poses either by simple scan to scan or scan to map matching technique.

The majority of the research work in the domain of SLAM focuses on using laser scanner observations to generate an accurate geometric model of the environment. In addition to measuring the distance to an object, a typical laser scanner also quantifies the remission values, i.e. received optical power, after reflection from the surface. This remission value is termed as *intensity* and depends (among other parameters) on an intrinsic surface property (surface reflectivity) as well as extrinsic parameters such as distance to the surface and angle of incidence with respect to the surface normal. Hence theoretically speaking given a model that defines the influence of the extrinsic parameters, it is possible to acquire a pose-invariant measure of surface reflectivity which can serve as additional information in a wide variety of robotic applications. This chapter presents a *simple data-driven model* of laser intensities through which a pose-invariant measure of surface reflectivity can be acquired. In addition, this measure is used in an extension of the Hector SLAM [90] algorithm which employs it for robot pose estimation and furthermore augments geometric models of the environment with surface reflectivity characteristics. It is important to highlight the scope of the proposed approach within the SLAM framework. The proposed approach can serve as part of the front-end of SLAM algorithms that estimates the transformation (using surface reflectivity) between consecutive robot poses (edge constraints) and furthermore acquires a geometric model (occupancy grid) of the environment augmented with surface reflectivity characteristics. In principle, any graph SLAM back-end [57, 81, 82] can be coupled with the proposed approach as SLAM back-ends are considered to be sensor agnostic. The capability of acquiring a geometric model augmented with surface reflectivity characteristics provides the possibility of using this information in the context of global localization [35, 165] and loop closure [137, 149, 189, 213, 214]. To explain this briefly in a simplistic scenario, consider a robot that traverses an infinitely long corridor as shown in Figure 3.1. In this context if the robot pose estimation is based only on metric information, the uncertainty along the principle direction of the corridor is unbounded as shown in Figure 3.1(a). However, if the corridor contains surfaces of different reflectivity and the pose estimation algorithm uses this information then the robot pose can be determined accurately as shown in Figure 3.1(b). The emphasis of this chapter is on the development of a *data-driven model* of intensities and its usage in the *SLAM front-end*.



(a) Pose estimation using metric information

(b) Pose estimation with additional reflectivity information

Fig. 3.1: Comparison of uncertainty for pose estimation (without odometry estimates) with and without reflectivity information.

3.2 Related Work & Contribution

In the last few decades a large amount of research work has been carried out in the field of SLAM [59, 81, 82, 90] in which a robot generates a geometric model of its environment based on laser scanner observations. In contrast, the research work on the applications of laser intensities in the domain of SLAM and to a certain extent in the field of robotics is rather insignificant. In [61, 132], the authors use retro-reflective markers as artificial beacons due to their significant difference in surface reflectivity to identify landmarks for SLAM. The most relevant research work with respect to this chapter of the thesis is presented in [204] in which an iterative closest point (ICP) [9] variant is presented that uses intensities to determine point correspondences between consecutive scans for transformation estimation. The above mentioned approach makes the assumption that the robot pose does not change significantly thereby ignoring the influence of extrinsic parameters (distance to the surface and angle of incidence to the surface normal). In contrast to the approach mentioned above, this chapter focuses on developing a *data-driven* approach to model the influence of extrinsic parameters on laser intensities to acquire a pose-invariant measure of surface reflectivity. These reflectivity characteristics are stored in a *reflectivity map* (occupancy grid augmented with reflectivity characteristics) for which the pose-invariance property is important as the same surface might be observed by the robot from different poses. In addition, the reflectivity characteristics are also used for pose estimation by matching the current scan (equipped with intensities) with an already acquired reflectivity map. In this specific case the robot position can change significantly (depending on the map update rate).

In contrast to the field of SLAM, laser intensities have been used for localization [101] and visual odometry [116]. A brief comparison of the proposed approach with those in the above mentioned domains is carried out to highlight the differences. In [101], the authors propose an approach based on particle filters that use intensities to localize an autonomous car in a highway scenario within an apriori known map. The approach uses lane marking for localization which are highly reflective in nature in comparison to asphalt to aid driving thereby not requiring any extrinsic parameter correction. In contrast, this chapter focuses on an approach that explicitly models the influence of distance and angle of incidence on intensities. The work presented in [116] proposes an appearance based mechanisms (detecting SIFT/SURF features on a planar projection of 3D intensity point cloud) for LIDAR sensors to perform visual odometry. It is important to highlight that in context of visual odometry the focus is on pose estimation based on consecutive images (during which the pose does not change significantly) without constructing a map. As the pose of the robot does not change significantly between consecutive images, the effect of distance and angle of incidence can be ignored to a large extent (in the paper the authors model the effect of distance, however they ignore the influence of angle of incidence). In addition, the appearance based mechanisms (SIFT and SURF features) are not directly applicable to the 2D Hokuyo/SICK Lidars used in this chapter. In addition, laser intensities have also been used for human detection [22], terrain classification [199] and object tracking [47, 65].

The main contributions of this chapter are highlighted below:

- A *simple data-driven* approach to model laser intensities for different scanners (Section 3.3.2)
- An extension of Hector SLAM capable of acquiring geometric models augmented with surface reflectivity characteristics (Section 3.4)
- An evaluation of the proposed *data-driven* approach and Hector SLAM extension (Section 3.5)

3.3 Modeling Laser Intensities

This section is divided into two main subsections. The first subsection focuses on the motivation of developing a *data-driven* approach to model laser intensities whereas the second subsection discusses the details of this data-driven approach.

3.3.1 Motivation for a Data-driven Approach

This subsection discusses the intensity characteristics of the most commonly used scanners in the field of robotics namely Hokuyo UTM-30LX and SICK LMS 291-S05[1]. To identify the extrinsic parameters which influence the intensity characteristics, it is essential to consider the LIDAR equation which is commonly used in the field of remote sensing [75, 152]. The LIDAR equation given the lambertian reflector assumption defines the relation between the received optical power P_{rec} and extrinsic parameters

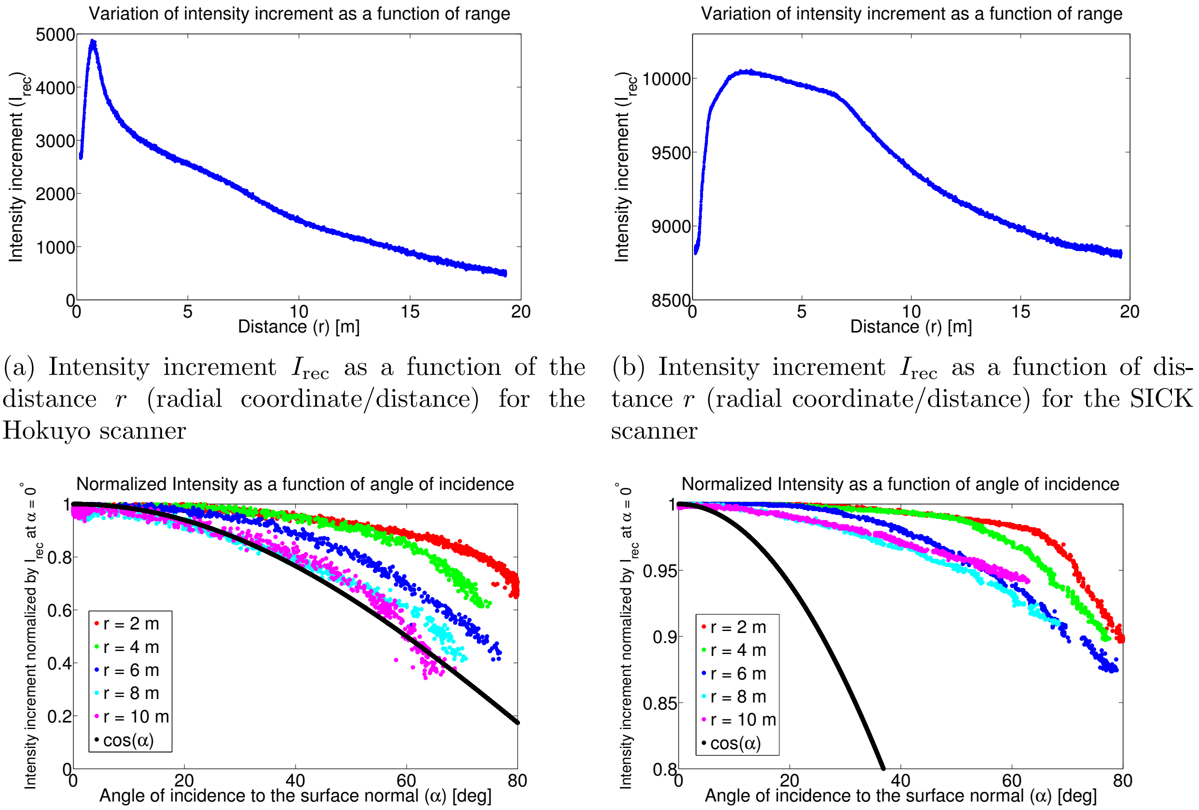
$$I_{\text{rec}} \propto P_{\text{rec}} \propto \frac{\varrho \cos(\alpha)}{r^2}, \tag{3.1}$$

where ϱ represents the surface reflectivity, r represents the distance (radial coordinate/distance) to the surface and α corresponds to the angle of incidence with respect to the surface normal. The proportionality between P_{rec} and extrinsic parameters exists due to presence of additional constant parameters such as

$$P_{\text{rec}} = \frac{P_{\text{emit}} d_{\text{apt}}^2 \varrho \cos(\alpha) \tau_{\text{sys}}}{4r^2},$$

the emitted power P_{emit}, system transmission factors τ_{sys}, aperture diameter d_{apt} etc. [44, 71]. I_{rec} represents the intensity increment, which is obtained after post-processing of the received optical power P_{rec} by the laser scanner. The intensity increment is assumed to be proportional to the received optical power. Eq (3.1) defines the parameters which influence intensities, hence the distance r and the angle of incidence α are the extrinsic factors that need to be considered during the modeling phase. In contrast, ϱ is an intrinsic surface property; which is useful for differentiating surfaces with different reflectivity properties. Although (3.1) contains all the extrinsic parameters that influence intensities, it is a crude approximation and does not consistently (over the complete domain of distance and angle of incidence) explain the empirical data for high-end terrestrial scanners [13, 44] as well as the laser scanners investigated in this thesis. To explain this briefly, consider the inverse square distance relationship in (3.1). Figure 3.2(a) and 3.2(b) shows the variation of the intensity increment I_{rec} for the Hokuyo and SICK scanner as a function of distance r (with a fixed angle of incidence $\alpha \approx 0°$) given the *same surface (fixed ϱ) i.e. standard white printing paper*. It can be seen that the inverse square distance relationship breaks down at close distances because I_{rec} starts decreasing instead of increasing. This effect has also been observed for high powered terrestrial laser scanners [13, 44] and has been termed the *near distance* effect. In photogrammetry and remote sensing literature this effect has been attributed to the defocusing of the receiver optics [44] (causing P_{rec} to decrease and consequently I_{rec} to decrease) for certain terrestrial laser scanners such as

[1] Intensities for the SICK LMS 291-S05 scanner were acquired by configuring the scanner to the undocumented measuring mode 13 (0Dh). The subcommand 2Bh can be used to request distance and intensity to which the scanner responds with the response F5h [51].



(a) Intensity increment I_{rec} as a function of the distance r (radial coordinate/distance) for the Hokuyo scanner

(b) Intensity increment I_{rec} as a function of distance r (radial coordinate/distance) for the SICK scanner

(c) Normalized intensity as function of angle of incidence α for different fixed distances r in case of the Hokuyo

(d) Normalized intensity as function of angle of incidence α for different fixed distances r in case of the SICK

Fig. 3.2: Intensity characteristics of the Hokuyo UTM-30LX and the SICK LMS 291-S05 scanner as a function of distance r in meters (radial coordinate/distance) and angle of incidence α in degrees with standard white printing paper as the surface that is being measured. a-b) The characteristics of the Hokuyo and the SICK scanner as function of distance r with a fixed angle of incidence to the surface normal ($\alpha \approx 0°$). Both scanner exhibit a decrease in intensity increment I_{rec} at close distances which is termed as the *near distance* effect [44]. The intensity characteristics are shown upto a distance of 19 meters as all the evaluations performed in this thesis were carried out indoors (18-20 m being the distance between the furthest surfaces). c-d) The variation in intensity as function of α given that the surface is observed at a fixed distance r in case of the Hokuyo and SICK scanner. The influence of the distance r is removed by normalizing the intensity, i.e. dividing the intensity increment with the value corresponding to $\alpha = 0°$, for a fixed distance r. Hence, the normalized intensity lies in the [0 1] interval. It is important to highlight that the angle of incidence is calculated by taking the dot product between the laser beam direction and the surface normal. The surface normal is the eigenvector corresponding to the smallest eigenvalue of the covariance matrix which is estimated by considering the neighbourhood around a certain point [158]. As the estimation of the surface normal degrades with point cloud density, the intensity characteristics could only be acquired upto $\alpha \leq 80°$ for small distances and $\alpha \leq 60°$ at large distances.

the Z+F[2] scanner. In principle, this effect is largely dependent on the intrinsic design and internal processing performed by the laser scanner (Riegl[3] scanners exhibit different intensity characteristics at near distances [13]), the details of which are not readily provided by companies making it difficult to ascribe a specific reason in case of the Hokuyo and SICK scanner. Similarly, in our evaluation the variation of normalized intensity as a function of α (after removal of the influence of r - see caption of Figure 3.2) also does not follow the $\cos\alpha$ model as shown in Figure 3.2(c) and 3.2(d). This inconsistency is generally attributed to the assumption that the surface should exhibit lambertian reflectance which is rarely the case. The highlighted inconsistency as well as the scarcity of system-based-models, due to lack of information from laser companies about the internal processing and intrinsic design, is the main motivation for developing a *data-driven* approach to model intensities. The objective of this model is to quantify the variation of intensity as a function of r and α to acquire a pose-invariant measure of surface reflectivity. Two different strategies can be adopted to develop a *simple data-driven* model, firstly assuming that the variation in intensity due to r and α can be modeled independently

$$I_{\text{rec}} \propto P_{\text{rec}} \propto \varrho f(r) f(\alpha), \tag{3.2}$$

where $f(r)$ and $f(\alpha)$ are the estimated data-driven functions defining the effect on intensities. In contrast, the second strategy is to develop a model

$$I_{\text{rec}} \propto P_{\text{rec}} \propto \varrho f(r, \alpha), \tag{3.3}$$

where $f(r, \alpha)$ jointly models the variation in intensities due to r and α. Figure 3.2(c) and 3.2(d) helps in assessing the plausibility of the assumptions in (3.2) and (3.3). If the assumption in (3.2) is true, the variation in the normalized intensity (effectively the removal of the influence due to r) should be the same at different r, however Figure 3.2(c) and 3.2(d) shows that this assumption does not hold for the Hokuyo and the SICK scanner at $\alpha \geq 20^\circ$ for different r. Given the trend in Figure 3.2(c) and 3.2(d), this thesis focuses on a *data-driven* approach to model intensities using (3.3).

3.3.2 Proposed Calibration Approach

This section defines a simple *data-driven* approach to model laser intensities and acquire a measure of surface reflectivity. Given a material with a known reflectivity coefficient ϱ, it is possible to calibrate and determine the function $f(r, \alpha)$ in (3.3). In case of unavailability of a surface with known reflectivity it is possible to acquire a *relative measure of surface reflectivity*. In this thesis the second option is considered due to its simplicity and applicability even in case of absence of standard materials with known reflectivity. Hence, the calibration process requires a *reference* surface (standard white printing paper) for which the intensities are measured as

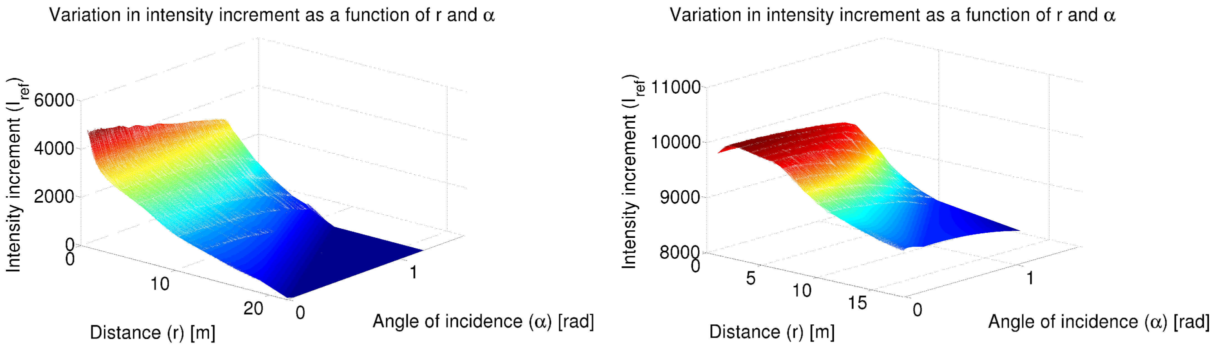
$$I_{\text{ref}} \propto P_{\text{ref}} \propto \varrho_{\text{ref}} f(r, \alpha). \tag{3.4}$$

[2] http://www.zf-laser.com/

[3] http://www.riegl.com/

Eq. (3.3) defines the intensity increment I_{rec} for a specific surface with reflectivity ϱ being currently observed at a specific r and α whereas (3.4) defines the intensity increment I_{ref} for the reference surface at the same r and α. Hence, (3.3) and (3.4) can be used to acquire a *relative measure of surface reflectivity* as

$$\frac{I_{\text{rec}}}{I_{\text{ref}}} \propto \frac{P_{\text{rec}}}{P_{\text{ref}}} \propto \frac{\varrho f(r,\alpha)}{\varrho_{\text{ref}} f(r,\alpha)} = \frac{\varrho}{\varrho_{\text{ref}}} = \bar{\varrho}. \tag{3.5}$$



(a) Approximated intensity increment $I_{\text{ref}} \propto \varrho_{\text{ref}} f(r,\alpha)$ for Hokuyo UTM30-LX

(b) Approximated intensity increment $I_{\text{ref}} \propto \varrho_{\text{ref}} f(r,\alpha)$ for SICK LMS 291-S05

Fig. 3.3: The approximated intensity increment $I_{\text{ref}} \propto \varrho_{\text{ref}} f(r,\alpha)$ surface of the Hokuyo and the SICK scanner obtained by using a scattered interpolant. This surface is furthermore sampled using a fine grid over r and α to generate a Lookup table (LUT) based model. As mentioned earlier the intensity characteristics are collected upto a distance of 18-20 m as all the evaluations were carried out indoors (for indoor scenarios this calibration is sufficient). If required the proposed approach can be applied in the same manner to acquire intensity characteristics over a wider r and α domain.

The relative measure $\bar{\varrho}$ defines the reflectivity of the measured surface with respect to the reference surface (white paper). *It is important to specify that this model assumes that the function $f(r,\alpha)$ varies in the same manner for all surfaces*, hence ignoring any coupling of the function f with ϱ. In the experimental evaluation carried out in indoor environments (see Section 3.5) this assumption yielded good results. The proposed approach, i.e. using the function $f(r,\alpha)$, is a data-driven formulation in contrast to the standard $\cos\alpha$ and inverse squared distance model. An important aspect of the proposed model is the approximation of I_{ref}. This approximation is performed by collecting observations of the reference surface at different r and α. Since it is not possible to acquire values at every r and α, a scattered interpolant (with linear interpolation) is used to approximate the values between given observations. This approximated surface obtained for the Hokuyo and SICK scanner is shown in Figure 3.3. This surface is furthermore sampled using a *fine* grid over r and α to generate a *lookup table* (LUT) based model. The main advantage of this LUT based model is that it can be computed offline and during online operation it requires simple array indexing thereby reducing computational cost.

3.4 Extension of Hector SLAM

This section focuses on using the relative reflectivity measure acquired in the previous section in an extension of the Hector SLAM [90] algorithm in which a robot acquires a geometric model augmented with a measure of surface reflectivity. The first subsection explains the occupancy grid structure whereas the second subsection focuses on the transformation estimation process based on the surface reflectivity measure by matching the current scan at time index t with an already acquired reflectivity map until time $t-1$.

3.4.1 Occupancy and Reflectivity Grid Structure

Let $G = \{g_1, \ldots, g_p\}$ represent the regular grid structure which stores two attributes, firstly the occupancy probability $P(g_i)$ and the surface reflectivity characteristics $R(g_i)$ observed for the i^{th} grid cell g_i. Let $z_t = \Big\{\{\mathbf{s}_1^t, \bar{\varrho}_1^t\}, \ldots \{\mathbf{s}_n^t, \bar{\varrho}_n^t\}\Big\}$ be the observation of the scanner at time index t consisting of n cartesian coordinates and surface reflectivity measures (obtained from the LUT based model correction). The notation $\mathbf{s}_i^t = [s_{i,x}^t, s_{i,y}^t]$ corresponds to the world coordinate scan end points. The occupancy probability of a grid cell is calculated using the standard recursive occupancy update equation defined in (2.6) [73, 89, 211]

$$P(g_i|z_{1:t}) = \left[1 + \frac{1-P(g_i|z_t)}{P(g_i|z_t)} \frac{1-P(g_i|z_{1:t-1})}{P(g_i|z_{1:t-1})} \frac{P(g_i)}{1-P(g_i)}\right]^{-1}.$$

The equation above can be converted to the log odds form to simplify the computation. In addition to the occupancy probability, the grid structure also stores the relative reflectivity characteristics of the surface (acquired from the LUT based model) for the i^{th} cell g_i. In the ideal case the reflectivity measure would be invariant to the robot pose thereby yielding a constant value for a specific surface, however a violation of the assumption in Section 3.3.2 or inaccurate surface normal estimation can cause reflectivity characteristics to vary. The reflectivity measure of each grid cell is calculated using a simple incremental averaging mechanism

$$R_{\mathrm{m}}(g_i|z_t) = R_{\mathrm{m}}(g_i|z_{t-1}) + \frac{{}^{i}\bar{\varrho}_j^t - R_{\mathrm{m}}(g_i|z_{t-1})}{n_{g_i}},$$

where $R_{\mathrm{m}}(g_i|z_t)$ represents the incremental mean of all the surface reflectivity observations till time index t. ${}^{i}\bar{\varrho}_j^t$ represents the j^{th} reflectivity measure in the sensor observation z_t for the i^{th} grid cell g_i and n_{g_i} represents the total number sensor observations for g_i. The left superscript of the reflectivity measure $\bar{\varrho}$ is not mentioned explicitly unless necessary for clarification.

Due to the discrete nature of the grid a bilinear interpolation scheme is adopted to allow subgrid accuracy as done in the original Hector SLAM paper [90]. However, the proposed approach interpolates the *relative surface reflectivity measure* rather than the occupancy probabilities and additionally frames the transformation estimation problem over this measure as discussed in the next subsection. Given a continuous coordinate P, the reflectivity characteristic $R(P)$ is approximated by using the four closest grid cells coordinates (assuming the indices to be (i, j, k, l) with $x_i = x_k$, $x_j = x_l$, $y_i = y_j$ and $y_k = y_l$. x_*, y_* are the metric coordinates of cell g_* in the geometric map) as

$$
\begin{aligned}
R(P) \approx \frac{y - y_i}{y_k - y_i} &\left(\frac{x - x_i}{x_j - x_i} R_{\mathrm{m}}(g_i) + \frac{x_j - x}{x_j - x_i} R_{\mathrm{m}}(g_j) \right) \\
+ \frac{y_k - y}{y_k - y_i} &\left(\frac{x - x_i}{x_j - x_i} R_{\mathrm{m}}(g_k) + \frac{x_j - x}{x_j - x_i} R_{\mathrm{m}}(g_l) \right).
\end{aligned}
$$

Similarly the gradient $\nabla R(P) = \left(\frac{\partial}{\partial x} R(P), \frac{\partial}{\partial y} R(P) \right)$ is approximated as in [90] by replacing the occupancy probabilities with the reflectivity measure

$$
\begin{aligned}
\frac{\partial R(P)}{\partial x} &\approx \frac{y - y_i}{y_k - y_i} \Big(R_{\mathrm{m}}(g_l) - R_{\mathrm{m}}(g_k) \Big) + \frac{y_k - y}{y_k - y_i} \Big(R_{\mathrm{m}}(g_j) - R_{\mathrm{m}}(g_i) \Big), \\
\frac{\partial R(P)}{\partial y} &\approx \frac{x - x_i}{x_j - x_i} \Big(R_{\mathrm{m}}(g_l) - R_{\mathrm{m}}(g_j) \Big) + \frac{x_j - x}{x_j - x_i} \Big(R_{\mathrm{m}}(g_k) - R_{\mathrm{m}}(g_i) \Big).
\end{aligned}
$$

3.4.2 Scan Matching

This section explains the robot pose estimation process for aligning new sensor observations with an existing reflectivity map. The proposed Hector SLAM extension formulates the estimation of the robot pose $\boldsymbol{\zeta} = \left[t_x, t_y, \theta \right]$ as the minimization of the cost function

$$
\boldsymbol{\zeta}^* = \arg\min_{\boldsymbol{\zeta}} \sum_{i=1}^{n} \left[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta})) \right]^2, \tag{3.6}
$$

where $\bar{\varrho}_i^t$ represents the reflectivity measure of the i^{th} beam end point in the sensor observation z_t and $R(\mathbf{S}_i(\boldsymbol{\zeta}))$ corresponds to the reflectivity measure in the map based on the transformed beam end point coordinates $\mathbf{S}_i(\boldsymbol{\zeta})$ as

$$
\mathbf{S}_i(\boldsymbol{\zeta}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} s_{i,x}^t \\ s_{i,y}^t \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \tag{3.7}
$$

Given an initial pose estimate of the robot, the objective is to find $\Delta\boldsymbol{\zeta}$ which minimizes the error

$$
\sum_{i=1}^{n} \left[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta} + \Delta\boldsymbol{\zeta})) \right]^2 \to 0. \tag{3.8}
$$

Using the first order Taylor series expansion of $R(\mathbf{S}_i(\zeta + \Delta\zeta))$ the expression becomes

$$
\sum_{i=1}^{n} \left[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta})) - \nabla R(\mathbf{S}_i(\boldsymbol{\zeta})) \frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} \Delta\boldsymbol{\zeta} \right]^2 \to 0.
$$

Taking the partial derivative w.r.t $\Delta\boldsymbol{\zeta}$ and setting it to zero

$$2\sum_{i=1}^{n}\Big[-\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Big]^T\Big[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta})) - \nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Delta\boldsymbol{\zeta}\Big] = 0,$$

$$\sum_{i=1}^{n}\Big[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Big]^T\Big[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta})) - \nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Delta\boldsymbol{\zeta}\Big] = 0.$$

By rearranging the terms, the above equation can be written as

$$\sum_{i=1}^{n}\Big[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Big]^T\Big[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Delta\boldsymbol{\zeta}\Big] = \sum_{i=1}^{n}\Big[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\Big]^T\Big[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta}))\Big]. \tag{3.9}$$

Solving (3.9) for $\Delta\boldsymbol{\zeta}$ yields the Gauss-Newton equation

$$\Delta\boldsymbol{\zeta} = \sum_{i=1}^{n}\mathbf{H}^{-1}\left[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right]^T\big[\bar{\varrho}_i^t - R(\mathbf{S}_i(\boldsymbol{\zeta}))\big],$$

where $\mathbf{H}$ corresponds to the hessian matrix which is calculated as

$$\mathbf{H} = \left[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right]^T\left[\nabla R(\mathbf{S}_i(\boldsymbol{\zeta}))\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}}\right].$$

The term $\frac{\partial}{\partial \boldsymbol{\zeta}}\mathbf{S}_i(\boldsymbol{\zeta})$ can be easily calculated from (3.7) as

$$\frac{\partial \mathbf{S}_i(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} = \begin{pmatrix} 1 & 0 & -\sin(\theta)s_{i,x}^t - \cos(\theta)s_{i,y}^t \\ 0 & 1 & \cos(\theta)s_{i,x}^t - \sin(\theta)s_{i,y}^t \end{pmatrix}.$$

In addition, the proposed extension of Hector SLAM takes advantage of the multi-resolution map as in [90] to escape local minima. An advantage of framing the pose estimation problem on gradient based methods is that the pose uncertainty can be directly computed from the inverse of the hessian matrix $\mathbf{H}$ as

$$\mathbf{K} = \sigma^2\mathbf{H}^{-1}$$

where $\mathbf{K}$ is the approximated covariance matrix and σ is a factor dependent on the sensor properties. This uncertainty can furthermore be used by SLAM back-ends [81, 82] to estimate the posterior distribution over the complete pose graph.

3.5 Experimental Evaluation

This section presents a quantitative evaluation of the proposed approach. The first subsection focuses on highlighting the importance of the LUT model by showing the effect of ignoring the influence of extrinsic parameters whereas the second subsection presents an evaluation of the proposed Hector SLAM extension.

3.5.1 Evaluation of the LUT Model

To highlight the advantage of the proposed approach it is important to consider alternative models that ignore the influence of extrinsic parameters (r and α). The following subsection gives a brief description of the alternative models considered in this thesis for comparison with the proposed approach.

Alternative Models

Given the extrinsic parameters (r and α) two different possibilities can be considered, firstly a model which ignores the effect of both r and α and directly uses the intensity increment I_{rec}. From here on in, this model is titled the *raw model*.

The second possibility is to model the influence of r, however systematically ignore the influence of α. Hence, this model corrects the intensity increment I_{rec} based on $f(r)$ which is generated by fitting a polynomial

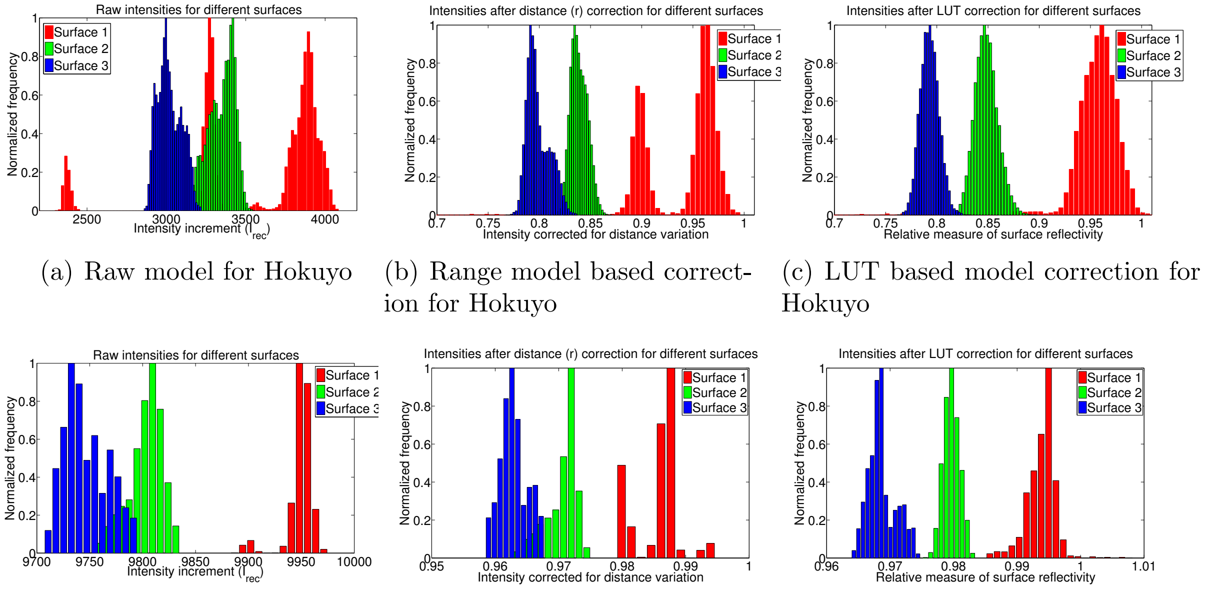
$$f(r) = \sum_{i=1}^{n+1} p_i r^{n+1-i},$$

to the intensity increment curve shown in Figure 3.2(a) and 3.2(b). Normalizing the intensity increment I_{rec} by the reference (white paper) polynomial curve $f(r)$ corrects the sensor observation based on r, however ignores the influence of α. This model is titled the *range model* for further reference.

Quantitative Evaluation

To highlight the importance of extrinsic parameter correction (r and α) and the ability of the LUT in differentiating between surfaces of different reflectivities, a quantitative evaluation is performed in comparison to the alternative models. To acquire data for this quantitative evaluation, *the laser scanner is mounted in a push-broom configuration (scanning vertically while the robot moves horizontally) thereby acquiring 3D models* of the environment as shown in Figure 3.8.

From the point cloud data, different samples (36000 point observations in total) were collected from the 3 different surfaces marked in Figure 3.8(a). The points sampled from surface 1 correspond to different extrinsic parameters (r and α) whereas the sampled points of surface 2 and 3 exhibit significant variation in α only. Figure 3.4 shows the histograms after applying different models (raw, range and LUT based model) for the Hokuyo and SICK scanner. Considering the Hokuyo scanner first (see Figure 3.4(a), 3.4(b) and 3.4(c)), it can be seen in Figure 3.4(a) that the raw intensity histograms of surface 2 and 3 exhibit overlap whereas the histogram of surface 1 is multimodal. Applying the *range* model, it can be seen in Figure 3.4(b) that the histogram of surface 1 exhibits bimodality due to α variation whereas the histograms of surface 2 and 3 still overlap. Figure 3.4(c) shows the proposed approach (LUT based model correction) in which the histogram of surface



(a) Raw model for Hokuyo

(b) Range model based correction for Hokuyo

(c) LUT based model correction for Hokuyo

(d) Raw model for SICK

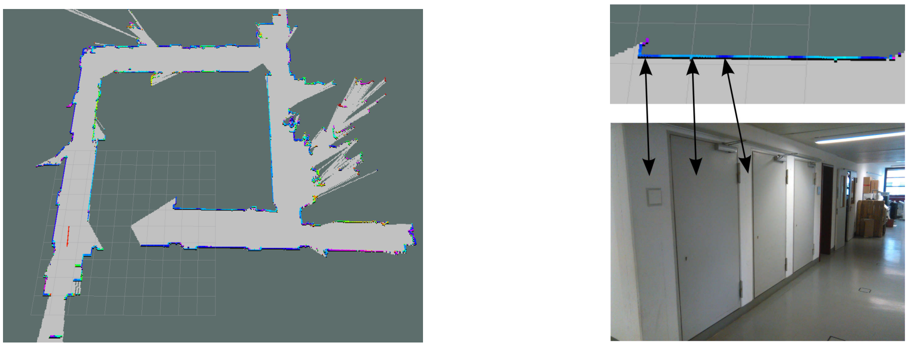
(e) Range model based correction for SICK

(f) LUT based model correction for SICK

Fig. 3.4: The histogram of intensities (with and without any correction) for different samples acquired from three different surfaces (see Figure 3.8(a)). The samples acquired from Surface 1 differ in r and α whereas the samples of surface 2 and 3 only vary in α. a,b) The histogram of intensities for the raw and the range model (see Section 3.5.1). The histograms corresponding to the raw and range correction model exhibit multimodality for surface 1 (due to r and α variation respectively) whereas surface 2 and 3 overlap. c) In contrast the proposed LUT based model ($\bar{\varrho}$) shows unimodal histograms for all three surface, hence it is capable of identifying that these surfaces have different reflectivity characteristics. d,e) The histogram of intensities for the raw and range correction model for the SICK scanner. It can be seen that the histogram of surface 2 and 3 overlap. f) The LUT based model is capable removing the overlap between the histograms of surface 2 and 3 and makes all three histograms identifiable as surfaces of different reflectivity characteristics.

1 becomes unimodal whereas the overlap between the histograms of surface 2 and 3 has been effectively removed.

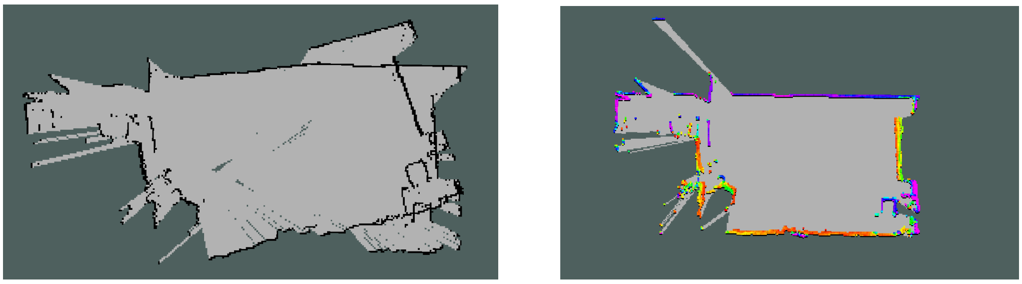
Figure 3.4(d), 3.4(e) and 3.4(f) show the same scenario in context of the SICK scanner. The first aspect to notice is that the variation in the intensity due to r and α is not as significant as in the case of the Hokuyo (see Figure 3.2). The histograms of surface 1 and 2 are separable even without extrinsic parameter correction whereas an overlap exists between the histograms of surface 2 and 3 due to variation in α. The *range* model shown in Figure 3.4(e) does not provide any significant advantage, however the *LUT* based model correction is capable of removing the overlap between the histograms of surface 2 and 3. *Hence, the evaluation of this section shows that extrinsic parameter correction is essential in context of identifying surfaces of different reflectivity characteristics.*



(a) Occupancy grid augmented with surface reflectivity characteristics (HSV map)

(b) Correspondance between reflectivity (HSV) map and actual surface

Fig. 3.5: a) Occupancy grid augmented with surface reflectivity characteristics (HSV colormap) acquired by the proposed extension of Hector SLAM. b) A zoomed in section of the occupancy grid of Figure 3.5(a) with the correspondences shown with the actual surface using arrows. *The laser scanner is mounted at a height of approximately 70 cm from the ground.* The corridor section visible in the color image is also observable in Figure 3.8(a).



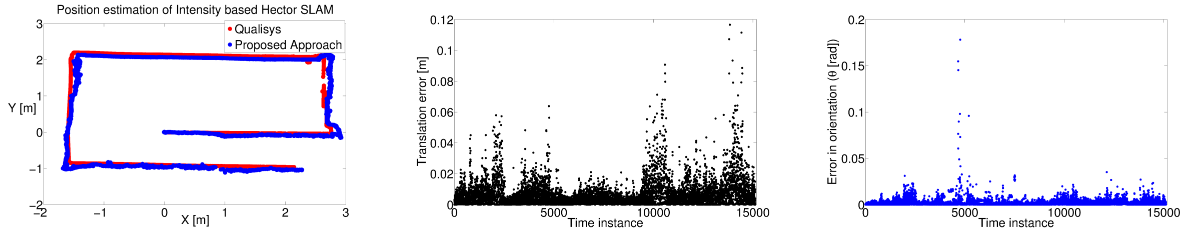
(a) Hector SLAM

(b) Intensity based Hector SLAM

Fig. 3.6: A specific scenario highlighting the advantage of intensity based Hector SLAM over standard Hector SLAM. The field of view (FOV) of the scanner based on the minimum and maximum angle is $[-1.047 \ 1.047]$ radians. The dimensions of the room are approximately 8.5m × 5.5m. a) Hector SLAM failed to create a consistent map as it could not find sufficient geometric features for pose estimation while turning at two different corners. b) Intensity based Hector SLAM succeeded in generating a consistent map as it additionally utilizes surface reflectivity for pose estimation.

3.5.2 Evaluation of the Hector SLAM (front-end) Extension

This subsection evaluates the proposed Hector SLAM extension. To present a concise evaluation and avoid repetition of similar conclusions/figures this section presents the results using the Hokuyo scanner, however the conclusions are valid for the SICK scanner as well. Figure 3.5(a) shows the reflectivity map of the corridor at the Chair of Automatic Control Engineering (shown with a HSV colormap) whereas Figure 3.5(b) shows one specific section of the occupancy grid marked with arrows to highlight the correspondence with the actual surface. In addition, Figure 3.6(a) and Figure 3.6(b) highlight the advantage of



(a) Robot trajectory in comparison to groundtruth

(b) Translation error over time

(c) Orientation error over time

Fig. 3.7: a) Comparison of the trajectory estimated by the robot using the proposed extension of Hector SLAM with groundtruth (from the Qualisys MOCAP system). b,c) A plot of the translation and orientation errors [19] showing that the proposed approach is capable of estimating the robot pose accurately.

intensity based Hector SLAM over standard Hector SLAM. In this specific scenario Hector SLAM failed to create a consistent metric map as it could not find sufficient geometric features for pose estimation while turning at two different corners. In contrast, intensity based Hector SLAM succeeded as it relied on surface reflectivity characteristics. In addition to the qualitative results in Figures 3.5 and 3.6, a quantitative evaluation of the proposed Hector SLAM extension is carried out using the MOCAP (motion capture) data acquired from the Qualisys system[4] which is capable of measuring the robot position with millimeter accuracy. Figure 3.7(a) shows the visualization of the ground truth trajectory (Qualisys system) as well as the robot positions obtained from the Hector SLAM extension. It is important to specify that qualisys motion capture system requires coverage (via external cameras) over the complete region where the robot has to be tracked, hence the evaluation of the motion could not be carried out in a large area. Figure 3.5(a) and 3.7(a) show that the proposed relative reflectivity measure can be used effectively to estimate the robot pose. In addition, a quantitative evaluation of the error for the proposed approach is performed using the metric defined in [19]

$$\epsilon(\delta) = \frac{1}{N}\sum_{ij}(\delta_{ij} \ominus \delta^*_{ij})^2,$$

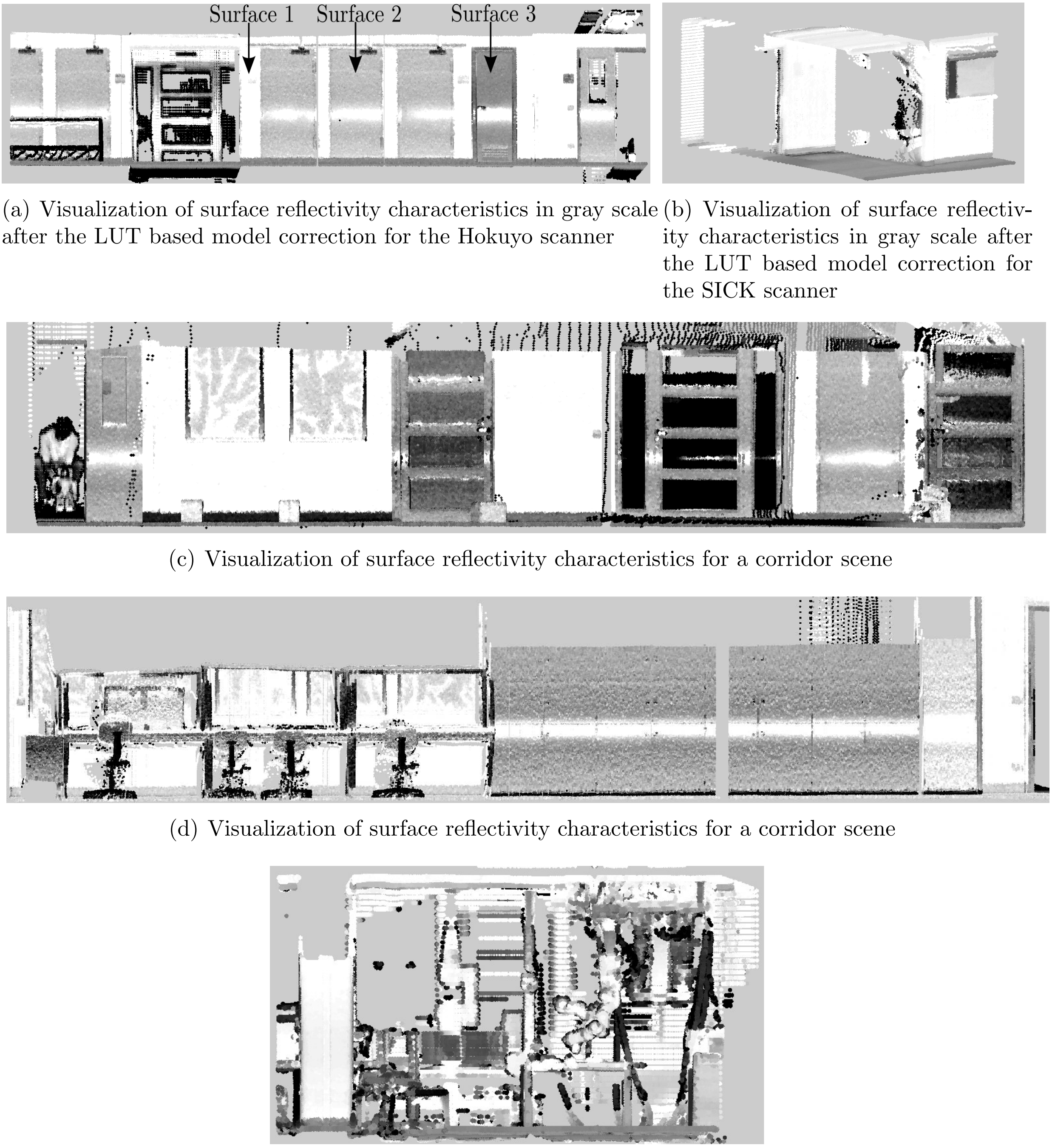
where δ_{ij} corresponds to the difference between consecutive robot poses at time index i, j and δ^*_{ij} corresponds to the ground truth variation in the pose. This δ_{ij} difference is split into the translation and the orientation error which is shown separately as a function of time in Figure 3.7(b) and 3.7(c) as in [19]. Figure 3.7(b) and 3.7(c) show that the magnitude of the delta translation and orientation error of the proposed approach is quite low. *Hence, the evaluation of this section highlights that the proposed approach is capable of estimating the robot pose accurately as well as acquiring a geometric model augmented with surface reflectivity characteristics.*

[4] http://www.qualisys.com/

3.6 Conclusion and Futurework

The domain of SLAM allows a robot to create a map in an online, incremental manner by coupling the pose estimation process with any given form of environment representation. The aspect of environment representation, specifically variable resolution grid based environment representation, was the focus of the previous chapter of this thesis. This chapter contributes in the domain of SLAM by discussing a *data-driven* approach to model laser intensities and identifies its role for pose estimation and grid based environment representation. The main purpose of modeling laser intensities is that they are dependent on the surface reflectivity (intrinsic parameter) as well as additional extrinsic parameters such as distance and angle of incidence to the surface. Thus by modeling the influence of extrinsic parameters, it is possible to acquire a measure of surface reflectivity which can be added as additional information in the map of the environment. An evaluation of the proposed data-driven approach is carried out in indoor environments to highlight the effects of ignoring the influence of extrinsic parameters when acquiring a measure of surface reflectivity from laser intensities. In addition, an extension of Hector SLAM is presented which uses this reflectivity measure for pose estimation and environment representation thereby acquiring a reflectivity map of the environment in an online, incremental fashion. The experimental evaluation highlights that the proposed extension possess the capability of acquiring an accurate robot pose estimate as well as a reflectivity map which can be useful for a wide variety of robotic applications.

Future work includes an evaluation of the relative reflectivity measure in outdoor urban environments under challenging weather conditions i.e. rain or snow. It will also be interesting to look into scenarios where the intensity based Hector SLAM approach can fail i.e. cases in which the normal vector estimation is inaccurate for a majority of the sensor observations due to low point density. In such cases it would be beneficial to combine surface attributes (reflectivity/color) with metric information along the lines of [74, 78].



(a) Visualization of surface reflectivity characteristics in gray scale after the LUT based model correction for the Hokuyo scanner

(b) Visualization of surface reflectivity characteristics in gray scale after the LUT based model correction for the SICK scanner

(c) Visualization of surface reflectivity characteristics for a corridor scene

(d) Visualization of surface reflectivity characteristics for a corridor scene

(e) Visualization of Kuka lab

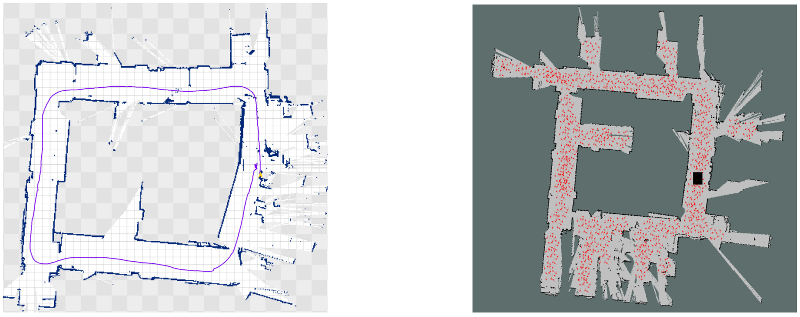
Fig. 3.8: a-c) Visualization of surface reflectivity characteristics in gray scale image after the LUT based model correction ($\bar{\varrho}$) with an additional linear scaling step to enhance contrast. A substantial region of the intensity point cloud shown in a) is also visible in the color image of Figure 3.5(b). It is important to highlight that the white horizontal region visible in a,c) across different surfaces is present due to specular reflection (in contrast to the standard diffuse reflection). This specular reflection occurs due to shiny and smooth surfaces as a significant amount of the emitted power is reflected back from the surface causing the receiver to register a maximum reading.

4 Appearance based Place Recognition/Loop Closure Detection

Summary and Contribution: *This chapter focuses on the aspect of visual appearance based place recognition/loop closure detection in the field of mobile robotics. The contribution of this chapter is twofolds: firstly the proposal of an online, incremental mechanism for binary vocabulary generation in the domain of loop closure detection. The second contribution is to evaluate the advantage of laser intensities for place recognition under challenging lighting conditions using different features and projection models. An extensive experimental evaluation is carried out to highlight the advantage of the proposed binary vocabulary generation mechanism as well as the usage of laser intensities for place recognition.*

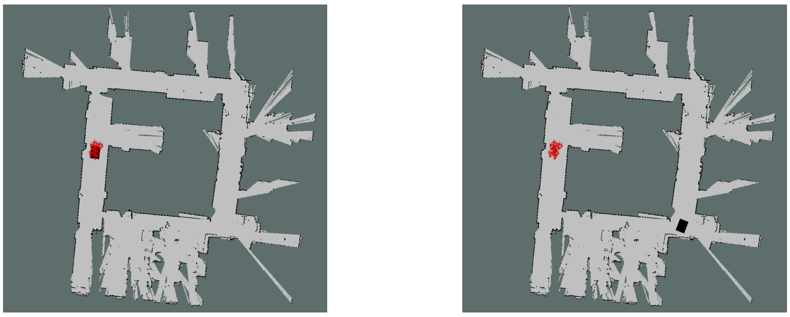
4.1 Introduction

The problem of *place recognition* plays an important role in different fields such as computer vision and robotics. The previous chapter of this thesis focused on the domain of SLAM, which allows a robot to generate a map in an online, incremental manner. This chapter focuses on the aspect of loop closure/place recognition problem within SLAM that allows a robot to maintain the consistency of the map over time by recognizing previously visited places and thereby reducing the error accumulated in the robot poses (see Figure 4.1(a) which highlights the error in the absence of a loop closure/place recognition algorithm). The most generic form of the place recognition problem can be found in the computer vision community in which (typically) given an observed image and unordered samples of images from discrete locations i.e. a database of images, the objective is to find a correspondence between the observed image and the database using a specific similarity metric. In the field of robotics, the place recognition problem plays a vital role in the domain of SLAM, localization and consequently navigation. The problem of *place recognition* with an additional temporal consistency constraint over sensor observations is titled the loop closure problem [3, 137, 138] in SLAM. The loop closure detection mechanism is a component of the graph SLAM front-end that generates edge constraints between nodes once the robot returns to a previously visited location. An effective performance of the loop closure detection mechanism is important for SLAM as a single incorrect loop closure constraint (edge constraint) can produce an inconsistent map. The importance of an accurate loop closure detection mechanism is further enhanced by the fact that most SLAM back-ends do not filter the generated edge constraints for consistency and leave this up to the front-end. To develop truly autonomous robots that are capable of generating consistent maps, loop closure mechanisms should work at 100% precision while maintaining high recall rate. Figure 4.1(a) shows a simple scenario in which a robot returns to a previously visited location, however due to its inability to close the loop it generates an inconsistent map. The groundtruth map consists of a corridor with all corners at right angles to each other. In SLAM, loop closure detection is required only once to correct the map, however this is just a functional requirement and there is no constraint on the loop closure mechanism to stop recognizing places as the robot traverses previously visited locations in the map.



(a) (b)

Fig. 4.1: a) Loop closure detection failure causes an inconsistent map. The actual map is a corridor in which all corners are at right angles to each other. b) Global localization using particle filters (particles shown as red arrows). The amount of particles required increases significantly with an increase in the area mapped by the robot. A robust place recognition algorithm can resolve this problem and reduce complexity.



(a) (b)

Fig. 4.2: a) An illustration of a localization algorithm that has converged at time instance t. b) The robot (shown as a black box) is kidnapped at the next time instance $t + 1$ and teleported to a different location. Most localization algorithms try to solve the global localization problem again as shown in Figure 4.1(b).

In context of robotic *localization* an interesting manifestation of the place recognition problem occurs during the initialization phase (global localization) [35] of the algorithm. In the initialization phase the localization algorithm does not have any prior distribution on the robot pose. In case there is no possible mechanism to determine a distribution over the robot pose the localization problem becomes quite challenging. A common solution to this problem is to use a particle filter to specify a uniform distribution over robot poses in the entire map, however this can be computationally expensive as shown in Figure 4.1(b). In principle this problem can be solved by extracting discriminative features from *passive* sensors (cameras) and use them to resolve the ambiguity. The example discussed above highlights the importance of developing robust place recognition algorithms that are capable of reducing ambiguity and providing an initial distribution over the robot position in the map. Another interesting case in context of localization is the kidnapped robot problem [42, 46] in which a robot is teleported to another location as shown in Figure 4.2. It is important to highlight the difference between loop closure, localization, global localization and the kidnapped robot problem. The difference between the localization and loop closure problem is quite subtle. In context of localization there is an implicit assumption that all observations are generated from a previously observed map whereas in the loop closure problem the map is incrementally being updated and the algorithm has to decide if an observation is generated from the prior observed map or if it is a new observation. In case of localization (with odometry) the initial robot pose is (generally) assumed to be known and the uncertainty is always bounded by the accuracy of the odometry estimates at all times. In context of the global localization problem there is unbounded uncertainty at time $t = 0$ (initially) which is reduced as sensor observations are obtained and eventually becomes bounded by the accuracy of the odometry estimates. In the kidnapped robot problem there is a possibility of unbounded uncertainty at all times t as the robot might be kidnapped at every (or any) time instance. Although the kidnapped robot problem is an imaginary construct (in reality a scenario in which a robot having a certain mass is kidnapped is not very likely), however it serves as an important benchmark to assess the reliability and robustness of a place recognition algorithm. In addition, the kidnapped robot problem has the effect of removing the prior over the robot position (possibly at every time instance) therefore transforming the problem into a generic form typically addressed within the computer vision community in which images/point clouds are retrieved from databases using a similarity metric. The discussion above provides a brief glimpse of the importance of place recognition in the domain of *robotics* as well as *computer vision*. The following paragraph describes a generic place recognition pipeline as well as it's important constituents.

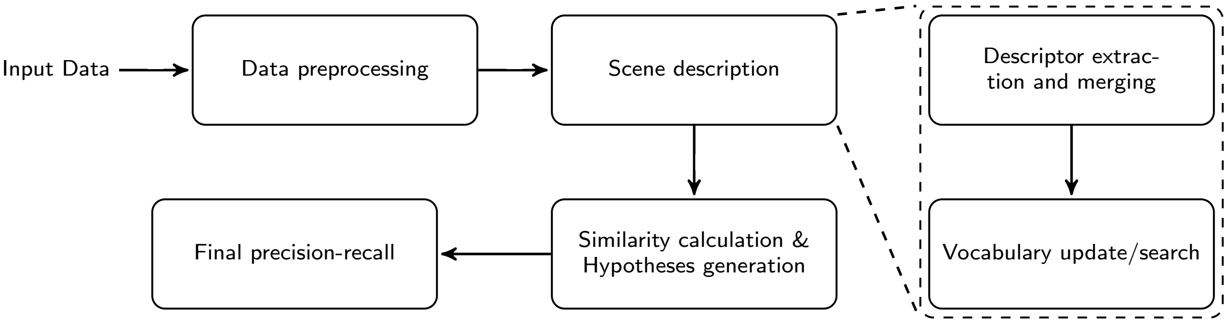


Fig. 4.3: The generic pipeline showing the set of operations performed on the input data for loop closure/place recognition.

Figure 4.3 shows a typical place recognition pipeline and its components. The input to the pipeline is the sensor data which can be a set of images/point clouds acquired from a camera or a laser scanner. This input data is further pre-processed e.g. converting color images to gray scale, downsampling images or point cloud or using the point cloud to generate images using different projection models (planar, equirectangular). The next component is the scene description block which summarizes the image using visual features (local or global) and updates/searches the vocabulary based on the new sensor observation. Furthermore this vocabulary is used to calculate the similarity between the current input and previous sensor data stored in the vocabulary. This similarity value is used to extract place recognition hypotheses, which can be used to determine the final candidate given certain constraints (such as temporal consistency). Furthermore, the pipeline can be evaluated in terms of its performance by using the output candidate to determine the precision-recall of the algorithm.

In the last few decades, a large amount of research has been carried out in the domain of place recognition. Although significant progress has been made however the state-of-the-art still faces challenges in real world scenarios. These challenges can be classified into two categories specifically *extrinsic* or *intrinsic*. Extrinsic challenges occur due to variations in the structure of the environment. The main extrinsic challenge for place recognition algorithms operating in typical outdoor scenarios with *passive sensors* (such as cameras) is the change in the environment appearance due to variations in ambient lighting. Even during different times of the day, shadows can cause a change in the environment appearance and pose challenges for place recognition algorithms [27]. In contrast to the *extrinsic* challenges mentioned above, intrinsic challenges correspond to the lack of information or capabilities that influence the operation of the place recognition algorithm. Examples of intrinsic challenges include deficiency of prior information available to the algorithm such as the lack of motion estimates (odometry) or the unavailability of GPS. In addition intrinsic challenges might also include the deficiency of prior training data for generating a visual vocabulary, which is typically the case in online robotic and computer vision applications. The extrinsic and intrinsic aspects mentioned above form a substantial set of challenges faced by place recognition algorithms in the field of robotics as well as computer vision. Although solutions to the place recognition problem have improved over time, however they still lack essential characteristics for robust operation in outdoor urban environments. Given the magnitude of issues highlighted above, an *ideal set of characteristics of a place recognition algorithm* are described below:

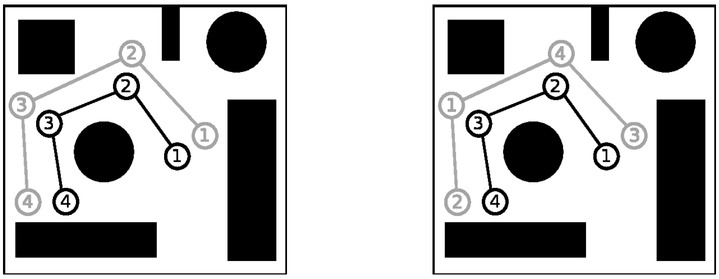
1. Capability of operating under adverse lighting conditions

2. Capacity of functioning in an online, incremental manner in case of unavailability of prior training data for vocabulary generation

3. Capacity of operating in the absence of *odometry*, *GPS* or *temporal consistency constraints* over sensor observations

4. Capability of generating high precision-recall

It is important to point out that the term *capacity* is used above to emphasize that these characteristics might not be a *strict necessity* depending on the application scenario in the field of robotics or computer vision. The first characteristic is essential to allow



(a) (b)

Fig. 4.4: An exemplary illustration of the temporal consistency assumption over sensor observations using a topological representation in which robot positions are represented by nodes with the integer inside the nodes representing the temporal sequence (1 represents t_1) (an observation is also associated with each node). The topological graph in black shows the first robot visit and the topological graph in gray represents the revisit. a) Due to the temporal consistency constraint the environment is visited in the same temporal sequence as during the first visit. b) The removal of temporal consistency allows the revisit to be performed by the robot in a random order, hence the robot can effectively *jump* or be kidnapped to another place in the state space. The removal of the temporal consistency constraint serves as a useful mechanism to assess the robustness and reliability of the place recognition algorithm. *It is assumed that a proper mechanism for topological map generation exists i.e. nodes are initializated after a fixed distance based on sensor characteristics or a keyframe (node) selection method exists, hence addition of nodes (between existing nodes) creates redundancy.*

operation in outdoor urban environments as it involves dealing with the variations in ambient lighting conditions. In general, the second aspect should be part of an ideal place recognition algorithm as it might not always be possible to have access to a large prior training dataset under varying lighting conditions for generating a visual vocabulary. The above mentioned scenario occurs specifically in the context of online robotic and computer vision applications. The third characteristic of an *ideal* place recognition algorithm is the capacity to function properly in case of unavailability of odometry, GPS or any temporal consistency constraint over observations. The removal of odometry, GPS or *temporal consistency constraint* over sensor observations serves as an effective test to determine if a place recognition algorithm can recover from the kidnapped robot problem. In general, most place recognition algorithms in robotic applications make an implicit assumption that the robot follows a certain trajectory and the sensor observations are in a temporal sequence corresponding to this trajectory. The removal of this temporal consistency constraint means that the algorithm can be presented with a random permutation of the temporal observations and it will still be able recognize similar places. Figure 4.4 shows an example to explain the scenario described above. As mentioned earlier, the removal of the temporal consistency constraint to address the kidnapped robot problem serves as a useful mechanism to assess the robustness and reliability of a place recognition algorithm. Another perspective of viewing the removal of temporal consistency constraint is to consider the generic place recognition problem in the domain of computer vision with the objective of retrieving similar images/point clouds from a database. The final aspect of the above mentioned characteristics is an *essential* requirement for all place recognition algorithms i.e. to generate high precision and recall as it highlights the reliability of the algorithm. The fulfillment of the above mentioned characteristics is a major challenge for place recognition algorithms.

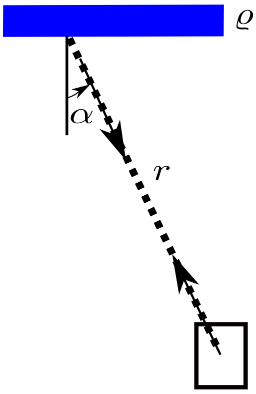


Fig. 4.5: The received optical power is dependent on an intrinsic surface property ϱ as well as extrinsic parameters such as distance r and the angle of incidence α to the surface normal.

This chapter is divided into two main parts. The first part of this chapter (see Section 4.4) focuses in the domain of loop closure and addresses the 2^{nd} and 4^{th} attribute of the ideal characteristics of a place recognition algorithm. An approach is presented that generates a binary vocabulary in an *online, incremental* fashion while maintaining high recall at 100% precision in comparison to the state-of-the-art. The proposed approach takes advantage of the temporal consistency constraint over sensor observations to generate loop closure candidates using visual appearance without requiring odometry or GPS information.

The second part of this chapter (see Section 4.5) addresses the place recognition from a general perspective and investigates the usage of laser intensities for place recognition given different pre-processing as well as scene description mechanisms. In contrast to passive sensors, laser scanners are capable of providing an intensity measure (back scattered energy from the surface) in addition to range data. Consider the LIDAR equation [44, 75, 153] based on the lambertian reflector assumption

$$P_{\text{rec}} \propto \frac{\varrho \cos(\alpha)}{r^2},$$

The equation states that the back scattered energy is dependent on an intrinsic property of the environment (surface reflectivity) and varies with distance as well as the angle of incidence to the surface. It is possible to calibrate the laser scanner and model the influence of the distance and angle of incidence to acquire a measure of surface reflectivity. Hence, the main advantage of using laser intensities is that they are invariant to ambient light sources and depend on an intrinsic property of the environment surface. An extensive evaluation of the proposed pipeline is carried out on a challenging outdoor urban environment in context of the kidnapped robot problem, i.e. without temporal consistency constraint, GPS or odometry information, to highlight the importance of laser intensities for place recognition.

4.2 Related Work

In this section the related work is assessed in terms of the ideal characteristics defined for a place recognition algorithm in Section 4.1. The approaches proposed in the literature are categorized based on the sensor type (*active* (laser) or *passive* (camera)), the *description* of the environment generated by them as well as the requirement of prior training data. The *description* of the environment generated by a place recognition algorithm can be based on *local* or *global/holistic* descriptors. *Local* descriptors use different distinct keypoints and the local neighbourhood around those keypoints to generate a compressed description/representation of the environment (such as SIFT [106] or SURF [6] for images and Fast point feature histograms (FPFH) [159], Normal aligned radial features (NARF) [177], Unique signatures of histograms for surface and texture description (SHOT) [188] for point clouds). In contrast *global* descriptors, such as GIST [144] or HOG [33], use the entire image to generate a holistic description of the environment.

Majority of the research work carried out in the field of place recognition/loop closure [3, 30, 49, 70, 84, 149, 206, 213, 214] has been based on *passive* (camera) sensors using *local* descriptors to generate a BOW (Bag of Words) representation to recognize places. Bag of words [29, 141, 169] is a structure that has been adopted from the field of language processing and information retrieval that allows the representation of an image as a vector by defining the presence or absence of a visual word. The visual words are obtained by clustering descriptors obtained from images after the features extraction process. The literature on appearance based place recognition can be divided into two categories based on the vocabulary generation process: i) Offline and ii) Online, incremental approaches. Among the approaches discussed above [30, 49, 70, 149] require an offline vocabulary generation phase using prior training data whereas [3, 83, 84, 206, 213, 214] generate a vocabulary incrementally without the need of training data. In [31], the authors' propose a probabilistic framework which incorporates co-occurrence probabilities between visual words using a Chow Liu tree [26] in an offline vocabulary generation process to perform appearance based loop closure detection. The approach can be considered as the de-facto standard for loop closure detection due to its robustness. In [92], the authors' present an approach that performs loop closure detection and visual odometry using a vocabulary tree generated offline to produce real time visual maps.In [3], a probabilistic frame work is presented that combines a vocabulary of SIFT features and color histograms for loop closure detection. The features are extracted from a single image and a geometric consistency test based on epipolar constraints is performed to validate loop closure hypotheses. In [83, 84], an approach is presented that incrementally generates a vocabulary using SIFT features matched over a sliding window of images. Recently an approach for place recognition using binary bag of words has been presented in [49]. It uses an offline vocabulary learning phase to generate a vocabulary tree consisting of binary visual words. Furthermore, it uses temporal and geometric consistency tests to filter loop closure candidates. In contrast to *local* descriptor based approaches mentioned above, a different spectrum of approaches rely on *global/holistic* representations [127, 168, 179] for place recognition while operating under favourable illumination conditions. In [127, 168] the authors use the holistic GIST descriptor [144, 190], whereas [179] computes a BRIEF descriptor [21] in a holistic manner on a downsampled image for place recognition.

A different line of research, using *passive* sensors, focuses on place recognition/loop closure under challenging lighting conditions using *local* and *global* descriptors. The approach presented in [76] uses local descriptor co-occurrence statistics under different conditions (morning, evening and night) to generate a vocabulary of visual words. In [77] an approach is presented that learns stable and discriminative local descriptors under different lighting conditions. The approach presented in [118] uses SAD (sum of absolute difference) on a subsampled version of the entire image (thus falls into the category of *global/holistic* descriptors). Furthermore, it applies a local contrast enhancement and additionally makes a simplistic constant velocity assumption between the matching scenes to determine visually similar places. The contrast enhancement between neighbouring images and the constant velocity assumption implicitly encodes the temporal consistency constraint. Additionally the approach performs pre-processing (cropping) on images to reduce the field of view of all images. The approach presented in [134, 135] is capable of recognizing places under seasonal changes by generating a data association graph with a fixed number of edges that encodes the vehicle speed and explicitly defines a temporal consistency constraint. To recognize similar places, a minimum cost flow is calculated on this data association graph. In [178] an approach is presented that first predicts the seasonal changes and then uses a standard place recognition approach to determine if a place has been visited before. Another interesting line of approach with *passive* sensors is illumination invariant imaging [108, 115] which removes the effect of sunlight and shadows from images by modeling the camera characteristics.

In contrast to *passive* sensors, the research in the domain of global place recognition using *active* sensors (LIDAR) in 3D outdoor urban environments has not been addressed that frequently. In [15, 16, 208] different approaches for keypoint selection and descriptor generation on point cloud data are presented for place recognition using local submaps within the Atlas framework [14]. Another focus within this domain has been to use range data to generate vocabularies or learn classifiers in an offline learning phase using prior training data. In [176], an approach based on NARF descriptors [177] is presented that generates a vocabulary from range images using prior training data to recognize places. In [55] features are extracted from laser range data and an ada-boost binary classifier is trained *offline* for place recognition. In addition to measuring the distance a typical LIDAR additionally measures the *back scattered energy* from the surface which is termed as intensity. In [101], 2D intensity maps are used for fine-scale localization whereas the global localization problem is approximately solved by relying on a differential GPS (D-GPS). The work presented in [146] uses a fusion of multiple features e.g. surface normal, reflectivity as well as SURF features extracted from images in a mutual information based approach for place recognition.

4.3 Motivation & Contribution

This chapter contributes towards two different aspects of the place recognition problem. Firstly, in the domain of loop closure by proposing an online, incremental binary vocabulary generation mechanism. Hence the first contribution focuses on the second characteristic, i.e. online, incremental binary vocabulary generation using *local descriptors*, of an ideal place recognition/loop closure algorithm defined in Section 4.1. The main advantage of using binary descriptors to generate a binary vocabulary is that they offer similar performance to real valued descriptors (such as SIFT and SURF) at reduced storage and computational costs [100]. A large amount of literature mentioned in the field of loop closure focuses on offline generation of visual vocabularies, hence are not suitable for online robotic applications. Although online vocabulary generation [3, 52] mechanisms such as incremental K-means exist, however, they are not well suited for binary spaces as they rely on the Euclidean distance metric and assume real valued descriptors which can be averaged [126]. In contrast, this chapter presents a *simple* approach for *online, incremental binary vocabulary* generation for loop closure detection. The incremental binary vocabulary generation process is based on feature tracking between consecutive frames thereby making it robot pose invariant and ideal for detecting loop closures in real world scenarios. Evaluation of the proposed incremental vocabulary generation process coupled with a simple similarity function and a temporal consistency constraint shows that it is capable of generating higher precision and recall in comparison to the state of the art on publicly available datasets.

The second contribution of this chapter lies within the domain of place recognition problem specifically in identifying the role of laser intensities for determining similar locations under challenging lighting conditions. The main advantage of using laser intensities is that they are invariant to ambient light sources and depend on an intrinsic property of the environment surface. *Hence, the objective is to highlight the advantages and applicability of laser intensities for place recognition in contrast to other forms of sensor data such as images from camera or geometry information from laser scanners. To the authors' best knowledge the role of laser intensities for place recognition under challenging lighting conditions i.e. recognizing the same place during day and night time has not been addressed in literature.* A generic pipeline for place recognition is presented that uses laser intensities to deal with ambient lighting conditions and does not require prior training data, odometry, GPS or temporal consistency constraints over sensor observations.

In summary, the main contributions of this chapter of the thesis are

- An *online, incremental* approach for generating a binary vocabulary for loop closure detection (Section 4.4)
- To highlight the advantages and applicability of laser intensities for place recognition under challenging lighting conditions in comparison to other forms of sensor data i.e. images from cameras or geometry information from laser scanners (Section 4.5)
- An extensive evaluation of the proposed vocabulary generation mechanism and the place recognition pipeline using laser intensities (Section 4.6)

4.4 Appearance based Loop Closure Detection using Passive Sensors

This section describes a simple approach for appearance based loop closure detection using a binary vocabulary which is generated in an online, incremental fashion by tracking features between consecutive images. In literature, the loop closure detection mechanism is part of the front-end of the graph SLAM [57, 82, 186] which deals with the raw sensor data and generates edge constraints between nodes once the robot returns to a previously visited location. An effective performance of the loop closure detection mechanism is important for SLAM as an incorrect edge constraint can produce an inconsistent map. To develop truly autonomous robots that are capable of generating consistent maps, loop closure mechanisms should work at 100% precision while maintaining high recall.

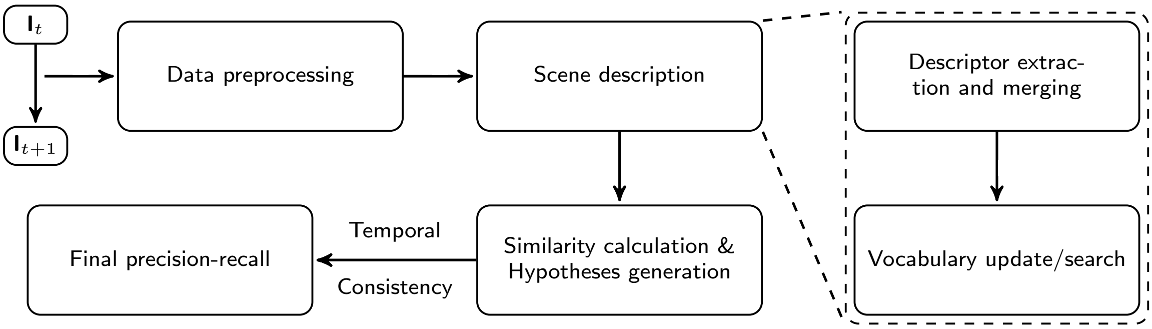


Fig. 4.6: The loop closure pipeline adapted from the generic pipeline shown in Figure 4.3.

The basic pipeline of operations performed on the images for detecting loop closures is shown in Figure 4.6. This pipeline is an adapted version of the generic pipeline shown in Figure 4.3 by enforcing temporal constraints on the sequence of input sensor observations as well as the output just before the final precision-recall calculation. The proposed approach takes advantage of the temporal consistency by matching features across consecutive images to acquire robot pose-invariant features. These features are then used to generate a binary vocabulary in an *online, incremental* manner without requiring any prior training data. The main contribution of this subsection is the *online, incremental* binary vocabulary generation mechanism which is a subcomponent of the *scene description* block. The proposed vocabulary generation mechanism is used to generate a hypotheses set of loop closures using a simple similarity function. This hypotheses set undergoes a consistency check by imposing a temporal consistency constraint over a larger horizon. The following subsections explain each component of the loop closure detection pipeline and the operations performed on the input data within those components.

4.4.1 Data Pre-processing

The loop closure detection pipeline operates over a consecutive pair of images. The data pre-processing step mainly performs RGB to gray scale conversion of the images received by the pipeline. These images are then passed onto the scene description component of the pipeline.

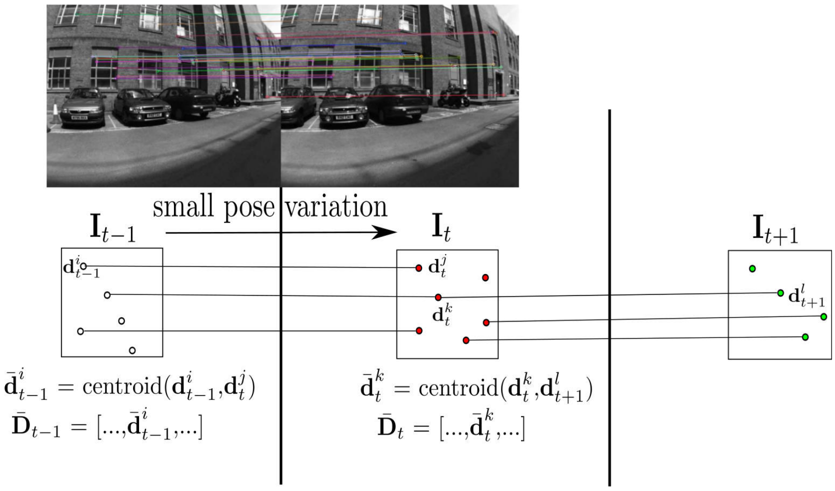

small pose variation
$\mathbf{I}_{t-1}$
$\mathbf{I}_t$
$\mathbf{I}_{t+1}$
$\mathbf{d}^i_{t-1}$
$\mathbf{d}^j_t$
$\mathbf{d}^k_t$
$\mathbf{d}^l_{t+1}$
$\bar{\mathbf{d}}^i_{t-1}$ = centroid($\mathbf{d}^i_{t-1}$,$\mathbf{d}^j_t$)
$\bar{\mathbf{D}}_{t-1}$ = [...,$\bar{\mathbf{d}}^i_{t-1}$,...]
$\bar{\mathbf{d}}^k_t$ = centroid($\mathbf{d}^k_t$,$\mathbf{d}^l_{t+1}$)
$\bar{\mathbf{D}}_t$ = [...,$\bar{\mathbf{d}}^k_t$,...]

Fig. 4.7: Descriptor extraction and matching mechanism between consecutive images to obtain view point invariant features. $\mathbf{d}^i_t$ represents the i^{th} descriptor extracted at time index t.

4.4.2 Scene Description

The scene description operates on the pre-processed images by extracting descriptors, merging (clustering) them and furthermore updates the vocabulary. In context of the loop closure pipeline discussed in this subsection, the scene description block only uses *local* descriptors i.e. local binary descriptors for vocabulary generation. The following subsections provides details on the different operations performed by each subcomponent of the scene description block i.e. descriptor extraction, merging (clustering) and vocabulary update.

Descriptor Extraction

The main steps carried out by this subcomponent of the pipeline is to match descriptors between consecutive images and extract view point invariant features. The proposed approach proposed uses BRISK (Binary Robust Invariant Scalable Keypoint) features, because they are scale and rotation invariant and offer similar performance to SIFT and SURF at reduced storage and computational costs [100].

The majority of the approaches [3, 31, 49] in appearance based loop closure rely on features extracted from a single image. In contrast, the proposed approach relies on matching features across consecutive images in a similar manner to [83, 214] and as shown in Figure 4.7. The purpose of matching descriptors across consecutive images (during which the robot undergoes slight variation in its pose) is to determine the most likely descriptors that will be observed in case the robot returns to the same location with a different pose. To match binary descriptors a metric has to be defined to measure similarity. In the proposed approach the Hamming distance is used which is defined as

$$H(\mathbf{d}_t, \mathbf{d}_{t+1}) = \sum_{i=1}^{p} (d_t[i] \veebar d_{t+1}[i]),$$

where $\veebar$ represents the *exclusive OR'* operator and p is the dimension of the descriptor vectors. The index i represents the i^{th} dimension of the p dimensional descriptor vector. $H(*,*)$ represents the Hamming distance whereas $\mathbf{d}_t$, $\mathbf{d}_{t+1}$ are the p dimensional descriptor vectors extracted from image $\mathbf{I}_t$, $\mathbf{I}_{t+1}$ respectively *at any keypoint* with t representing the time index. In effect, the descriptor matching process is an 'exclusive OR' between the bits of the descriptor vectors and a count of set bits along the entire descriptor dimension. Two descriptors matched across subsequent images are considered a good match if the Hamming distance between them is below the *matching threshold* δ whereas all descriptors which do not satisfy this threshold are discarded. The centroid of the matched descriptors is taken as their representative. The centroid $\bar{d}[i]$ of the i^{th} dimension of the binary descriptor vector at *any time index* is calculated as below

$$\forall i \leq p, \bar{d}[i] = \text{centroid}(d^1[i], d^2[i], ..., d^k[i])$$

$$= \begin{cases} 0 & \text{if} \sum_{j=1}^{k}(d^j[i]) < \dfrac{k}{2} \\ 1 & \text{if} \sum_{j=1}^{k}(d^j[i]) \geq \dfrac{k}{2} \end{cases}, \tag{4.1}$$

where the notation $d^j[i]$ represents the i^{th} dimension of the j^{th} descriptor vector and k represents the total number of descriptors whose centroid is being calculated. Equation 4.1 calculates the centroid for any arbitrary number of inputs k, however, in the proposed approach the centroid is calculated for descriptors matched during consecutive time indices as shown in Figure 4.7 and stored in $\bar{\mathbf{D}}_t$ at time index t.

Merging Descriptors

After the descriptor extraction process the next step involves merging descriptors with the objective of removing multiple instances of similar descriptors, as it might be the case that the image contains repetitive patterns. Let $\bar{\mathbf{D}}_t = [\bar{\mathbf{d}}_t^1, \bar{\mathbf{d}}_t^2, ..., \bar{\mathbf{d}}_t^m]^T$ (T and m denote the transpose and the total number of descriptors respectively) represent the centroid of descriptors matched between consecutive images $\mathbf{I}_t$ and $\mathbf{I}_{t+1}$. A descriptor after the merging process is termed as a visual word. The algorithm starts by matching a descriptor with all other descriptors in the set $\bar{\mathbf{D}}_t$. Descriptors are merged and replaced by their respective centroid in a *greedy* manner if the distance between them is below the matching threshold δ. This process continues until no further merging can take place. The psuedocode of the merging algorithm is shown in Figure 4.8. The descriptors after the merging are step are called visual words. The visual words obtained after merging, denoted $\hat{\mathbf{D}}_t$, are then used by the vocabulary to determine previous time instances when the same visual word was observed as well as the vocabulary update process.

Assignment of the BOW Index

The visual words $\hat{\mathbf{D}}_t$ obtained after merging/clustering are compared with the visual words present in the vocabulary denoted $\mathbf{V}_{t-1}$ (see Figure 4.10). This operation is performed to

```
Merge($\bar{\mathbf{D}}_t$)
Input: $\bar{\mathbf{D}}_t = [\bar{\mathbf{d}}_t^1, \bar{\mathbf{d}}_t^2, ..., \bar{\mathbf{d}}_t^m]^T$
          // Descriptors from feature extraction
Output: $\hat{\mathbf{D}}_t$ // Visual words

Procedure:
Initialize number of visual words to $m$;
  // $|\bar{\mathbf{D}}_t|$ represents the number of descriptors
    for-all ($i \leq |\bar{\mathbf{D}}_t|$)
       for-all ($j = i + 1$ till $j \leq |\bar{\mathbf{D}}_t|$)
             if ($H(\bar{\mathbf{d}}_t^i, \bar{\mathbf{d}}_t^j) < \delta$)
                $\hat{\mathbf{d}}_t^i$ = centroid($\bar{\mathbf{d}}_t^i, \bar{\mathbf{d}}_t^j$);
                $\bar{\mathbf{d}}_t^i = \hat{\mathbf{d}}_t^i$; //update descriptor for next iteration of $j$
                $\bar{\mathbf{D}}_t$– $\bar{\mathbf{d}}_t^j$; // remove $\bar{\mathbf{d}}_t^j$ from $\bar{\mathbf{D}}_t$
                decrement $m$;
                $\hat{\mathbf{D}}_t \longleftarrow \hat{\mathbf{d}}_t^i$; //copy/overwrite $i^{th}$ index in $\hat{\mathbf{D}}_t$
             endif;
        if merging not possible for iteration $i$
             $\hat{\mathbf{D}}_t \longleftarrow \bar{\mathbf{d}}_t^i$;
```

Fig. 4.8: The pseudocode of merging descriptors to remove multiple instances of binary descriptors in the same image

determine the number of the *new* and *old* visual words in $\hat{\mathbf{D}}_t$. This step is essential for the next block, i.e. similarity calculation and loop closure candidate selection process, to function properly. The matching threshold δ is used to match the descriptors in $\hat{\mathbf{D}}_t$ with the visual words in the vocabulary to determine the indices of the *old* visual words. The indices of all the *old* visual words are stored in the set S. The pseudocode of the above mentioned process is shown in Figure 4.9. An important point to mention here is that the vocabulary index $t-1$ is used here because the update based on the visual words detected in $\mathbf{I}_t$ occurs at the end of pipeline, hence after the similarity as well as the loop closure hypotheses calculations. Initially, at time $t = 0$ the vocabulary is empty, hence all visual words are initialized as *new* and stored in V_{-1}.

Vocabulary Structure

This subsection given an overview on the vocabulary structure used by the loop closure pipeline. The vocabulary is the most important component the loop closure pipeline. Besides storage of binary visual words in matrix $\mathbf{V}_{t-1}$, the vocabulary also contains:

- Occurrence frequency of all binary visual words
- Inverted index for generating loop closure hypotheses

The occurrence frequency denoted $\mathbf{F}_{t-1} = [f_{t-1}^1, f_{t-1}^2, ..., f_{t-1}^n]$ contains the number of times a specific visual word is observed in the images till time $t-1$. The term n represents

```
Assign-BOW-Index($\hat{\mathbf{D}}_t$, $\mathbf{V}_{t-1}$)
Input: $\hat{\mathbf{D}}_t$ // visual words
       $\mathbf{V}_{t-1}$ // visual words in vocabulary
                at time index $t-1$
Output: $S$ //set of indices of old visual words
              found in $\hat{\mathbf{D}}_t$
        $N_{new}$ // number of new visual words

Procedure:
  for-all ($i \leq |\hat{\mathbf{D}}_t|$)
    word_found = false;
    for-all ($j \leq |\mathbf{V}_{t-1}|$)
      if ($H(\hat{\mathbf{d}}_t^i, \mathbf{v}_{t-1}^j) < \delta$)
        $S \longleftarrow j$; // store visual word index
        word_found = true;
        break;
      endif;
    if ($\sim$ word_found)
      increment $N_{new}$;
    endif;
```

Fig. 4.9: The pseudocode of assigning merged descriptors a BOW index

the total number of visual words present in the vocabulary. The vocabulary also maintains an inverted index to generate loop closure hypotheses based on visual words detected in $\mathbf{I}_t$. In the proposed approach the inverted index is stored as a sparse binary matrix which describes the presence or absence of a visual word in all images till time index $t-1$ as shown in Figure 4.10.

4.4.3 Similarity Calculation and Loop closure Hypotheses

This subsection focuses on the similarity calculation and the hypotheses generation process for loop closure detection.

Loop Closure Hypotheses

Given the set S generated by the previous component of the pipeline, the loop closure hypotheses set can be generated by using the inverted index. As shown in Figure 4.10, given the indices of the old visual words detected in $\mathbf{I}_t$ and stored in the set S, their occurrence frequency and presence in previous images can be easily extracted. A *temporal constraint threshold* β (where $\beta > 0$) is used to prevent the algorithm from generating loop closure hypotheses with images observed close to the current time index. Hence, loop closure hypotheses are constrained to lie within the time index $t_i = 0$ and t_L. $t_i = 0$ represents the initial time index when the loop closure algorithm started and $t_L = t - \beta$, where t represents the current time index.

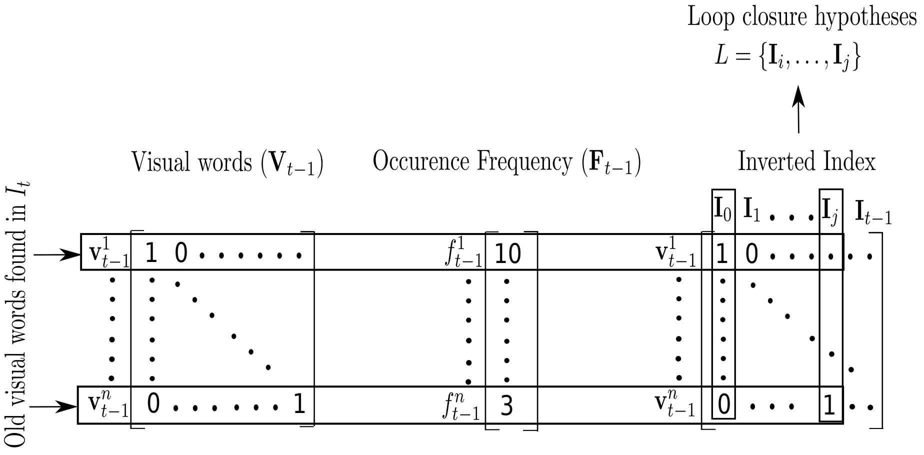

Loop closure hypotheses
$L = \{\mathbf{I}_i, \ldots, \mathbf{I}_j\}$
Visual words ($\mathbf{V}_{t-1}$)
Occurence Frequency ($\mathbf{F}_{t-1}$)
Inverted Index
Old visual words found in I_t
$\mathbf{I}_0$ $\mathbf{I}_1$ $\mathbf{I}_j$ $\mathbf{I}_{t-1}$
v^1_{t-1} 1 0
v^n_{t-1} 0 1
f^1_{t-1} 10
f^n_{t-1} 3

Fig. 4.10: An overview of the three main components of the vocabulary. The vocabulary consists of the binary visual words stored (row wise) in $\mathbf{V}_{t-1}$ ($n \times p$ matrix), the occurrence frequency of each visual word and an inverted index. The inverted index is stored as a sparse binary matrix representing the absence or presence of visual words in all images till time index $t-1$. Given the indices of the old visual words stored in S generated during the assignment of bag of word index, it is possible to determine past images containing the same visual words as shown above.

Similarity Calculation

Let $L = \{\mathbf{I}_i, ..., \mathbf{I}_j\}$ (where $i \geq 0$ and $j \leq t_L$) represent the set of loop closure hypotheses generated from the inverted index and U represents the set of common visual words between loop closure hypothesis image $\mathbf{I}_i$ and currently observed image $\mathbf{I}_t$. The similarity of hypothesis $\mathbf{I}_i$ with the current image $\mathbf{I}_t$ is calculated as

$$\mathcal{S}(\mathbf{I}_i, \mathbf{I}_t) = \frac{\sum_{\forall m \leq |U|} (f^m_{t-1})^{-1} |U|}{\sum_{\forall m \leq |U|} (f^m_{t-1})^{-1} |U| + \sum_{\forall k \leq |T|} (f^k_{t-1})^{-1} |T| + N_{new}},$$

where T consists of indices of visual words (extracted from the inverted index) present in $\mathbf{I}_i$ but not found in $\mathbf{I}_t$. The notation $|T|$ and $|U|$ represents the cardinality of the set. f^j_{t-1} represents the occurrence frequency of the j^{th} visual word in the vocabulary. N_{new} is the number of new words detected in $\mathbf{I}_t$. The above mentioned similarity metric $\mathcal{S}$ is a modified version of the jaccard index. The normalized similarity of the loop closure candidates is calculated as

$$\hat{\mathcal{S}}(\mathbf{I}_i, \mathbf{I}_t) = \frac{\mathcal{S}(\mathbf{I}_i, \mathbf{I}_t)}{\sum_{\forall \mathbf{I} \in L} \mathcal{S}(\mathbf{I}, \mathbf{I}_t)},$$

where L as defined earlier is the entire hypotheses set. The final loop closure candidate is chosen as one that leads to the maximum value of the normalized similarity function.

$$\arg \max_{\forall \mathbf{I} \in L} \hat{\mathcal{S}}(\mathbf{I}, \mathbf{I}_t).$$

To prevent the algorithm from generating loop closure candidates based on a single hypothesis a constraint is placed that $|L| > 1$.

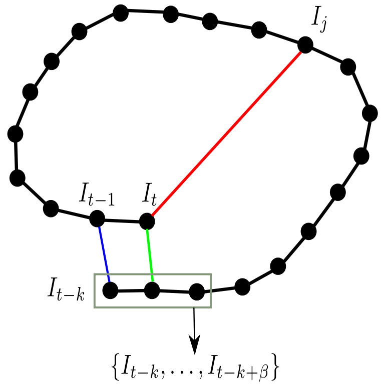


Fig. 4.11: (Best visualized in color) Given the accepted loop closure at time index $t-1$ (shown in blue), the loop closure at time index t is constrained to lie in $t-k$ till $t-k+\beta$ (shown in green). The loop closure with image $\mathbf{I}_j$ (shown in red) is rejected as it does not satisfy the temporal constraint.

Temporal Consistency

The loop closure candidate chosen in the last step of the pipeline goes through a simple temporal consistency test. The temporal consistency test is based on the time index of the previously observed loop closure. Consider a scenario in which a robot at time index $t-1$ returns to a location which was previously visited at time index $t-k$ where $k > \beta$. The temporal consistency test states that after the loop closure event between $\mathbf{I}_{t-1}$ and $\mathbf{I}_{t-k}$, all future loop closure events detected in the interval of t and $t+\beta$ are constrained to lie between $t-k$ and $t-k+\beta$. In Figure 4.11, it can be seen that due to the temporal consistency constraint given the loop closure event at time index $t-1$, the loop closure event between $\mathbf{I}_t$ and $\mathbf{I}_j$ is rejected (shown in red) whereas the loop closure event in the interval of $t-k$ till $t-k+\beta$ (shown in green) is accepted. In case the robot return to the same location multiple times in the past, the temporal consistency test has to be extended to all such time intervals.

Vocabulary Update

Once the similarity calculation and the loop closure hypothesis generation has been completed, the vocabulary is updated by expanding the vocabulary size based on the number of *new* visual words detected in I_t. Additionally, the occurrence frequency of all the old visual words has to be updated and for all the new visual words it has to be initialized to 1. Finally, the inverted index is updated based on the visual words detected in the current time index. After the vocabulary update, the loop closure mechanism waits for the next input image and then performs all the steps of the pipeline again from the beginning.

4.4.4 Precision-recall Calculation

The evaluation of the loop closure pipeline is performed using precision-recall curves. The precision of the algorithm is defined as the ratio of correct recognized places among the total number of loop closure events determined by the algorithm. The recall is the ratio of the number of correct detections with respect to the ground truth loop closure events.

4.5 Place Recognition using Passive and Active Sensors

The previous section discussed different aspects of the loop closure pipeline using passive sensors (images from cameras) with a focus on *online, incremental* binary vocabulary generation with temporal consistency constraints. This section addresses the place recognition pipeline from a general perspective and evaluates the discriminative abilities of different forms of sensor data (images from camera or intensities/geometry information from laser scanners) for place recognition without requiring any temporal consistency constraints, GPS or odometry. Figure 4.3 shows the generic pipeline and the set of operations performed on the input data such as point clouds, camera or range images. As shown in Figure 4.3, an initial preprocessing step (generating specific projections or downsampling) is performed on the input data depending on the overall objective of the pipeline as discussed in Section 4.5.1. After the preprocessing the description of the scene is calculated (shown as the scene description block in Figure 4.3) based on *local* or *global* descriptors. Given a specific similarity function, it is possible to calculate the similarity between different scenes and generate a square symmetric similarity matrix. This similarity matrix is then used to generate the hypotheses of places that look similar to each other as well as the final precision-recall curve. The rest of the section gives a detailed description of each component of the pipeline.

4.5.1 Data Pre-processing

The basic operations being carried out in the data preprocessing block is dependent on the type of sensor data being fed into the place recognition pipeline. If the input is 3D point clouds then the most common preprocessing step is to filter and downsample them to reduce computational cost. In contrast if the input is camera images then the standard preprocessing stage includes image resizing or conversion to gray scale. It is also possible to generate these images from point cloud data using specific projections (equirectangular or rectilinear). The advantage of generating a projection of the point cloud is that the problem that is initially posed over a 3D space is reduced to a 2D representation. Two different projections are considered in this chapter, specifically equirectangular and rectilinear/cubic projection. The main advantage of the equirectangular projection is the 360° field of view (panoramic image) which can be beneficial for place recognition algorithms. In contrast, the rectilinear/cubic projection has a limited field of view, however the advantage of this projection is that straight lines in the environment remain straight after the projection (almost all *local* and *global* descriptors are developed, optimized and evaluated for this specific projection). In addition, the rectilinear/cubic projection can be directly extracted from the equirectangular projection given the observer orientation [155, 172, 201]. The rectilinear/cubic projection is used to generate planar intensity images from laser scanner data to evaluate the performance of the place recognition algorithms in comparison to camera images. The notation $\mathbf{I}_i$ is used to represent an image (given any input type and projection model i.e. range or intensity image with equirectangular or rectilinear projection) generated by the projection of the i^{th} point cloud. The projection is highlighted by the upper right superscript i.e $\mathbf{I}_i^{\text{eqrect}}$, $\mathbf{I}_i^{\text{rect}}$ to represent equirectangular or rectilinear projection respectively. In addition, the image type is specified by the upper left superscript such as ${}^{\text{r}}\mathbf{I}_i$, ${}^{\text{i}}\mathbf{I}_i$ to define range and intensity images given any projection. Similarly, camera images, i.e. color images converted to gray scale images during pre-processing, are specified by the upper left superscript ${}^{\text{c}}\mathbf{I}$. The following subsections provide details about the projection models.

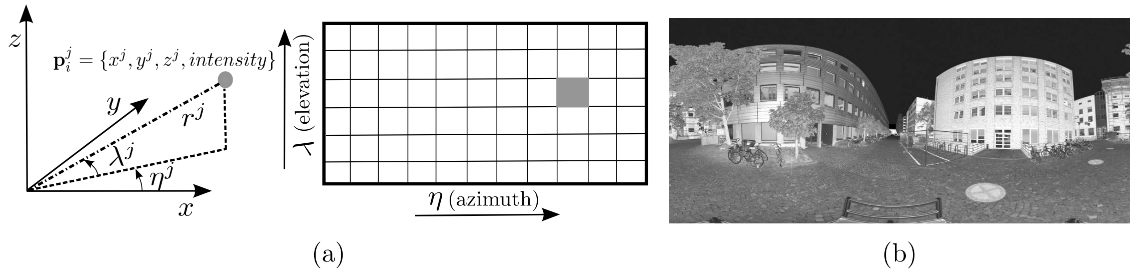


Fig. 4.12: (a) Laser scanner observation of the j^{th} point in the i^{th} point cloud $\mathbf{P}_i$. (b) Equirectangular intensity image obtained after projecting the point cloud. The azimuth and elevation of the j^{th} point is denoted by η^j and λ^j respectively.

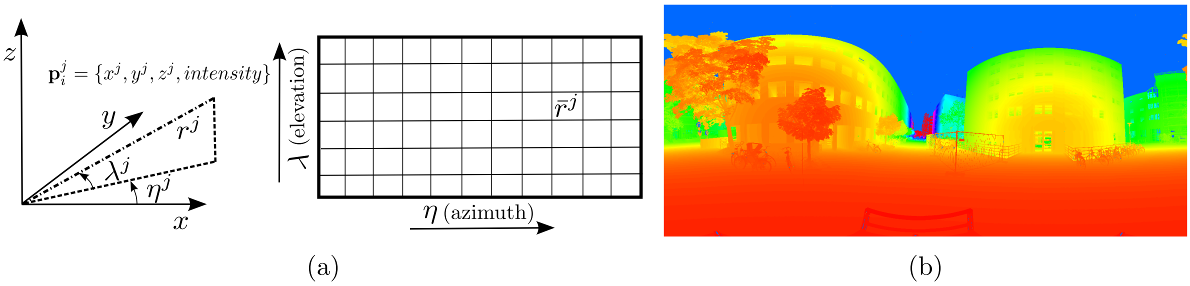


Fig. 4.13: (a) The process of range image generation in which the range value is accumulated in the relevant elevation, azimuth bin. Furthermore, this range image is normalized by the maximum range (as represented by $\bar{r}^j$ in the figure) to generate a matrix of floating point values between 0 and 1. b) (Best visualized in color) An example of the generated range image visualized with a HSV colormap.

Equirectangular Projection

The main advantage of the equirectangular projection is that it generates a panoramic image of the environment as shown in Figure 4.12. The equirectangular projection takes the azimuth and elevation of each point defined in spherical coordinates and interprets them as rows and columns of a matrix. Given the row, column index of the matrix the range or intensity value is accumulated in that specific bin leading to range or an intensity image. The pseudo-code for generating an equirectangular projection from a point cloud is defined in Appendix A.1.

Rectilinear/Cubic Projection

In contrast to the equirectangular projection, the rectilinear/cubic projection model is used to generate planar intensity images which are compared with camera images in terms of discriminative capabilities for place recognition using the same pipeline. The rectilinear projection can be obtained from the equirectangular projection as shown graphically in Figure 4.14. The pseudo-code for generating this projection is shown in Appendix A.2.

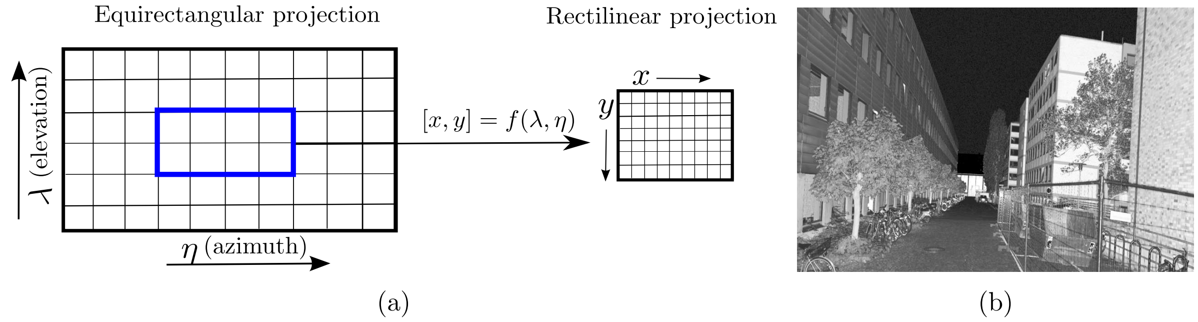


Fig. 4.14: a) An abstract representation of mapping the equirectangular coordinates to the rectilinear image coordinates. b) Front rectilinear projection corresponding to the image shown in Figure 4.12(b).

3D Point Cloud Pre-processing

In contrast to the previous subsections in which a projection of the point cloud is used to generate an image, it is also possible to utilize the point cloud directly for place recognition. A Bag of Words (BOW) approach similar to subsection 4.4.2 is presented which uses *local descriptors* extracted from point clouds. The main preprocessing step performed on the 3D point cloud is downsampling it with a voxel grid to effectively reduce the computational cost. The grid resolution used to downsample the point cloud is denoted by ψ (the notation has been defined here for further reference during the experimental evaluation).

4.5.2 Scene description and Similarity calculation

The scene description and similarity calculation block shown in Figure 4.3 form the core components of the place recognition pipeline. Figure 4.15 shows the components of the scene description block in the pipeline. As mentioned earlier that the place recognition literature has been dominated by two sets of approaches: mainly using *local* or *global/holistic* descriptors. The set of operations performed in the scene description and similarity calculation block differ for both approaches. The following subsection focuses on the scene description and similarity calculation block for *local* descriptors whereas the subsequent subsection focuses on the same components for *global/holistic* descriptors.

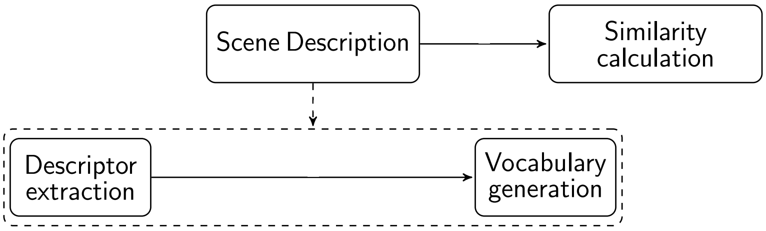


Fig. 4.15: The components of the scene description block in the pipeline. It is important to specify that the subcomponents of the scene description block (descriptor extraction and vocabulary generation) differ in the set of operations performed depending on the descriptor type (*local* or *global* descriptors).

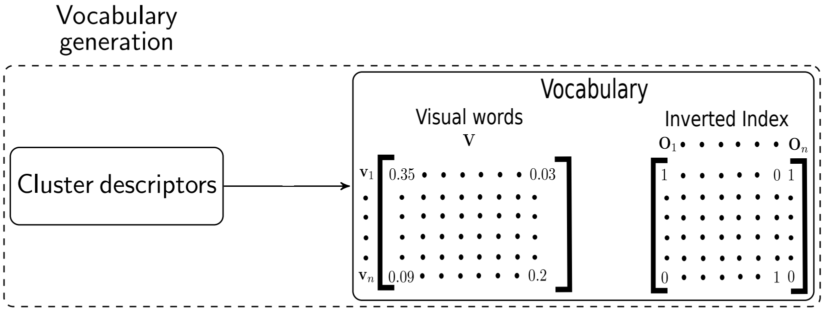


Fig. 4.16: The processes involved in the vocabulary generation block shown in Figure 4.15 for *local* descriptors. The visual words are stored row wise in the matrix **V**. The inverted index is binary matrix which is used as a look up table to determine which visual words occurred in a specific image/point cloud $\mathbf{O}_i$.

Local Descriptor based Vocabulary Generation (BOW approach) and Similarity Calculation

This section explains the bag of words approach using *local* descriptors for place recognition. The vocabulary generation process involves extracting descriptors (descriptor extraction block in Figure 4.15) from the input data which can be laser scanner images i.e. intensity (${}^{\mathrm{i}}\mathbf{I}$)/range (${}^{\mathrm{r}}\mathbf{I}$), camera images (${}^{\mathrm{c}}\mathbf{I}$) or 3D point clouds ($\mathbf{P}$). Let $\mathbf{O}_i$ be a generic notation to represent an image $\mathbf{I}_i$ or point cloud $\mathbf{P}_i$. SIFT [106] features have been used for the BOW place recognition pipeline with images as input. In context of BOW place recognition using 3D point clouds (based on local descriptors), the SHOT [188] descriptor has been used due to its robustness and repeatability [164]. Given the descriptors extracted from an image/point cloud $\mathbf{O}_i$, they can be used to incrementally generate a visual vocabulary. The vocabulary generation process is similar to the mechanism defined in Section 4.4.2 such that it uses a fixed distance δ to generate visual words in an incremental manner without any prior training data. As the vocabulary generation process is formulated over real valued descriptors (SIFT), the distance δ defines a ball (B_δ) in the Euclidean space around the descriptor. It is also possible to apply a similar formulation for binary descriptors such as BRIEF [21] and BRISK [100] using the hamming distance (in hamming space) to define δ as done in Section 4.4.2. The main processes involved in the vocabulary generation process include an initial clustering phase followed by a vocabulary update step. The purpose of clustering descriptors separately for all images/point clouds at the initial stage is to remove multiple instances of similar descriptors as it might be possible that they contain repetitive environment structure (e.g. multiple windows with same structure on a building). After the clustering process the vocabulary update process is carried out. The main components of the vocabulary includes a matrix of visual words $\mathbf{V}$ (stored row wise) and an inverted index which helps to determine which visual words were observed in different images/point clouds. The update process of the vocabulary involves determining the presence of a specific visual word and to accordingly update the inverted index. The basic components of the vocabulary are shown in Figure 4.16. Given a specific state of the vocabulary at any time instance the similarity function is used to determine which places look similar to each other. A detailed description of the above mentioned steps is provided below:

- **Clustering:**

The objective of the clustering process is to remove multiple instance of similar descriptors that exist in the image/point cloud as it might contain repetitive patterns. Let $\mathbf{D}_i = \{\mathbf{d}_i^1, \ldots, \mathbf{d}_i^T\}$ be the set of descriptors extracted from an image (intensity, range images from laser scanner or camera images)/point cloud $\mathbf{O}_i$. Given the descriptors, a symmetric distance matrix $(T \times T)$ is calculated which constitutes the Euclidean distance between all the descriptors. Given the symmetric distance matrix, all descriptors with distance less than δ are merged into a single descriptor by averaging those descriptors. This process is continued until no further merging can take place. The descriptors after the merging process are denoted *visual words*. Let $\hat{\mathbf{D}}_i = \{\mathbf{d}_i^1, \ldots, \mathbf{d}_i^M\}$ denote the final set of visual words obtained after merging descriptors from the image/point cloud $\mathbf{O}_i$ where $M \leq T$.

- **Assignment of Visual Words:**

The basic structure of the vocabulary used in this pipeline is shown in Figure 4.16. The components are the visual words $\mathbf{V}$ and an inverted index (a binary matrix) used as a look up table to determine which visual words were observed in each image/point cloud in previous time steps. Given the visual words $\hat{\mathbf{D}}_i$ obtained in the clustering step, the next step is to compare them with the visual words in the vocabulary $\mathbf{V}$ using the distance δ to determine which words have already been observed and those that are new as discussed in Section 4.4.2 [213]. In case a visual word has already been observed before, it is updated by averaging it with the matched visual word present in the vocabulary and updating the inverted index. In contrast if a visual word is not present in the vocabulary, a new visual word is added to the visual vocabulary $\mathbf{V}$ and the inverted index expanded accordingly.

- **Similarity Calculation:**

This subsection focuses on the similarity calculation block shown in Figure 4.15. Given the vocabulary components defined above, it is possible to calculate the similarity between different images/point clouds at any instance. In the proposed approach a variant of the term frequency- inverse document frequency (tf-idf) is used as a similarity function as it has been used extensively in literature for place recognition [3, 141, 169]. Let $\mathbf{O}_i$, $\mathbf{O}_j$

represent the images/point clouds whose similarity is to be evaluated. Let q represent the set of indices (within the vocabulary) of all the visual words present in the image/point cloud $\mathbf{O}_i$ and $\mathbf{O}_j$. Let set k represent the set of indices of the common visual words between the images/point clouds with $k[l]$ representing the l^{th} element of the set and the notation $|k|$ representing the cardinality of the set. Let $n_{q[l]}$ represent the number of images/point clouds which contain the visual word present at $q[l]$ index in the vocabulary (extracted from the inverted index) whereas N represent the total number of images/point clouds in the vocabulary. Given the above mentioned information, the similarity $\mathcal{S}_{local}$ between images/point clouds $\mathbf{O}_i$ and $\mathbf{O}_j$ is calculated as

$$\mathcal{S}_{local}(\mathbf{O}_i, \mathbf{O}_j) = \sum_{l=1}^{l \le |q|} \mathrm{tf}(q, k) \times \mathrm{idf}(N, n_{q[l]}), \tag{4.2}$$

$$\mathcal{S}_{local}(\mathbf{O}_i, \mathbf{O}_j) = \sum_{l=1}^{l \le |q|} \mathbf{1}(q[l] \in k) \times \log \frac{N}{n_{q[l]}}, \tag{4.3}$$

where the first term ($\mathrm{tf}(q, k)$ in Equation 4.2) is called the term frequency which is used as a binary weighting term (due to its simplicity). In simple words the weighting factor is an indicator function for all the common visual words between the image/point cloud $\mathbf{O}_i$ and $\mathbf{O}_j$. The inverse document frequency ($\mathrm{idf}(N, n_{q[l]})$ in Equation 4.2) is modeled as a logarithmic function of the ratio of the total number of images/point clouds and the number of images/point clouds which contain the common visual words. The objective of the inverse document frequency term is to down weight all commonly occurring visual words. Hence, using the equations described above the similarity between images/point clouds can easily be calculated.

Holistic/Global Descriptor Representations and Similarity Calculation

This subsection discusses the scene description and similarity calculation block shown in Figure 4.15 for *global/holistic* descriptors such as GIST and HOG (Histogram of Oriented Gradients [33]). In context of *global/holistic* representations the main operational component of the scene description block is the descriptor extraction process as the *vocabulary generation block only stores the descriptors of the images for evaluating the similarity.* The following paragraphs provide details on the descriptor extraction and the similarity calculation process for global descriptors.

- **HOG based Holistic Image Representation:**

In literature HOG descriptors have been successfully applied for people detection [33] and human action recognition [187]. The most common application involves training a linear SVM (requiring prior training data) to detect a person in an image or alternatively to classify an action being performed by the person.

In the proposed approach the HOG descriptor is calculated directly on the image $\mathbf{I}_i$ and used as a global descriptor to represent the image without any requirement of a classifier. The basic steps involved in the calculation are similar to the original HOG descriptor. To calculate the HOG descriptor the image is divided into cells which are then aggregated into larger spatial blocks. The gradient computations are performed on the entire image and a histogram is generated for each cell. The cells are aggregated into blocks which are normalized. The normalized descriptors for all the blocks are then concatenated into one global descriptor for the entire image. The basic steps involved in the calculation of the global HOG descriptor used are described below:

- Pre-processing with a Gaussian filter of a spatial window size σ
- Gradient computation over image $\mathbf{I}_i$
- Spatial and Orientation binning within cells of size L
- Normalization of histograms using L_2 norm for all cells in a block
- Concatenating features of all blocks to form the global descriptor

The first step is the smoothing process using a Gaussian filter with a spatial window size σ. The smoothed image is then used to compute the simple 1-D centered gradient ([-1 0 1]) as

$$e_i(x) = I_i(x+1, y) - I_i(x-1, y),$$
$$e_i(y) = I_i(x, y+1) - I_i(x, y-1),$$

where $e_i(x)$ and $e_i(y)$ corresponds to the horizontal and vertical gradients at position $(x,\ y)$ of the image $\mathbf{I}_i$. Given the gradients, the orientation (θ) and the magnitude (γ) can be easily calculated as $\theta_i(x,y) = atan(e_i(y)/e_i(x))$ and $\gamma_i(x,y) = \sqrt{e_i(x)^2 + e_i(y)^2}$ respectively. The calculated orientation is then accumulated in its orientation histogram bin with a weighting factor defined by the magnitude. The histograms of all cells within a block are appended and normalized using L_2 norm. Finally the histograms of all blocks are concatenated to form the *global descriptor* denoted $\mathbf{d}_i$ for the image $\mathbf{I}_i$. The global descriptor described above is used to determine the similarity between different images based on the similarity function defined later in this section.

- **GIST based Holistic Image Representation:**

The concept of capturing the *gist* of a scene is a generic concept that has been studied extensively in human visual perception and computer vision. The basic idea behind determining the gist of a image is to capture its essence. One such approach has been presented in [144] which has found application in place recognition as discussed in Section 4.2. A brief overview is provided here for clarity, for further details see [144]. The approach introduces the concept of a spatial envelope which defines the shape of the scene. The authors define different properties of the spatial envelope such as degree of *naturalness*, *openness*, *roughness*, *expansion* and *ruggedness* and discuss that they are sufficient to distinguish between natural landscapes and man made scenes (urban environments). The spatial envelope is estimated by considering the spatial distribution of spectral information (spectogram) in the image. Furthermore it is shown that the spectogram varies significantly for different categories (such as natural landscapes and man made scenes), hence it is possible to distinguish between them. The proposed place recognition pipeline uses the Gabor GIST method explained in [144, 190][1]. The above mentioned approach divides the image into a grid of a specific resolution depending on the number of blocks φ provided as an input. For each block the local regions of the image are convolved with Gabor gist filters tuned for different scales Ω and orientations per scale χ (symbols are defined here for the evaluation in the experimental section). The mean response of the filter for each window is concatenated into a descriptor denoted $\mathbf{d}_i$ for image $\mathbf{I}_i$. Given the descriptor it is possible to estimate the similarity between images.

- **Similarity Function for Global Descriptors:**

Given the HOG or GIST global descriptors for any pair of images, the next step is to calculate the similarity between them using the similarity calculation block shown in Figure 4.15. The similarity $\mathcal{S}_{global}$ between two images $\mathbf{I}_i$ and $\mathbf{I}_j$ (given any image type and projection model) with their HOG *or* GIST descriptor abbreviated as $\mathbf{d}_i$ and $\mathbf{d}_j$ is defined as:

$$\mathcal{S}_{global}(\mathbf{I}_i, \mathbf{I}_j) = \frac{\mathbf{d}_i^T \mathbf{d}_j}{\sqrt{(\mathbf{d}_i^T \mathbf{d}_i)(\mathbf{d}_j^T \mathbf{d}_j)}},$$

which is effectively the cosine distance and a commonly used metric for measuring similarity between descriptors [135, 139, 198].

4.5.3 Precision-Recall Calculation

Given the similarity function for local and global descriptors it is possible to generate a square symmetric similarity matrix which defines the similarity score between all the images. It is important to specify that if no assumption is made about the structure of this square symmetric similarity matrix, it is equivalent to the notion of removing any temporal consistency constraint on the sensor observations or prior information about the robot position. The above mentioned square similarity matrix is thresholded to generate the hypotheses of places which look similar to each other. The true and false positives are extracted from these hypotheses based on the groundtruth and then used to calculate the precision-recall of the algorithm.

4.6 Experimental Evaluation

The experimental evaluation is divided into two subsections. The first section focuses on the evaluation of the loop closure approach proposed in Section 4.4 using passive sensors. The second subsection focuses on the evaluation of the generic place recognition pipeline explained in Section 4.5 as well as highlighting the advantages of using laser intensities for place recognition.

[1] http://people.csail.mit.edu/torralba/code/spatialenvelope/

4.6.1 Loop Closure using Passive Sensors

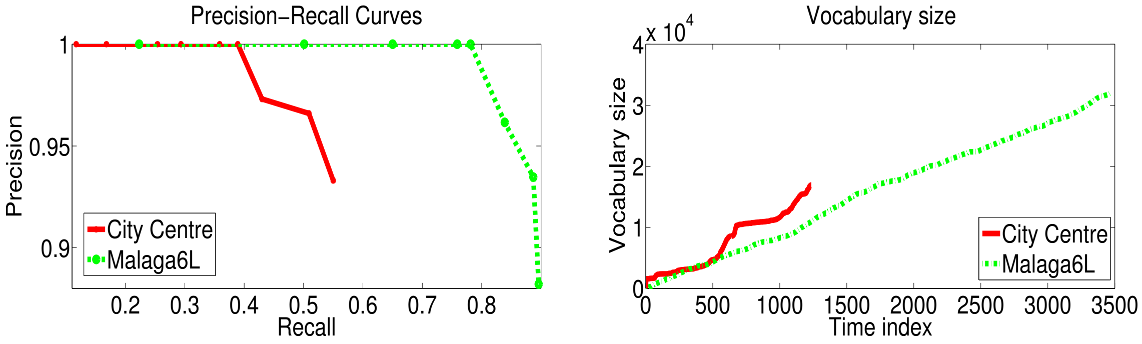
This section evaluates the performance of the proposed loop closure approach on different publicly available datasets, as shown in Table 4.1, by comparing it to state of the art methods such as FAB-MAP 2.0 [31] and the approach proposed by Gálvez-López [49] . For all dataset evaluations mentioned in this section the descriptor dimension p is 512 and the temporal constraint threshold β is set to 10. All experiments were performed on an Intel i5-2500K 3.3 GHz processor with 16 GB RAM.

Tab. 4.1: Details about the Datasets used in Evaluation

Dataset	Description	Camera position	Image size	# Images
Malaga6L [12]	Outdoor, slightly dynamic	Frontal	1024 x 768	3474
City Centre [30]	Outdoor, urban, dynamic	Lateral	640 x480	1237

Methodology

The correctness of the results for Malaga6L and City centre datasets is established by using the ground truth information and script used by the authors in [49] as a *black box*, hence without any modification in the parameters. The script determines the precision and recall of the algorithm given the ground truth information. The precision of an algorithm is defined as the ratio of correct loop closures to the total number of detected loop closures. The recall is the ratio of the number of correct detections to the ground truth loop closure events. The ground truth information (used in [49]) contains a manually created list of loop closures. *'The list is composed of time intervals, where each entry in the list encodes a query time interval associated with a matching interval'.*



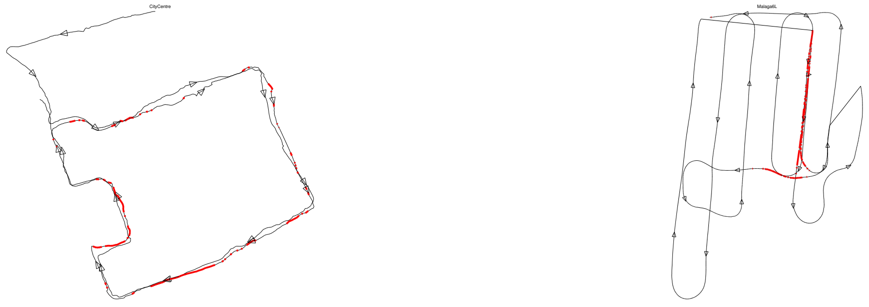
(a) Precision Recall Curves

(b) Vocabulary size

Fig. 4.17: (a) Precision recall curves of the proposed approach (b) Vocabulary size as a function of time for different datasets

Tab. 4.2: Results of Magala6L and City Centre dataset

Dataset	Approach	Precision (%)	Recall (%)
Malag6L	Gálvez-López [49]	100	74.75
	FAB-MAP 2.0 [31]	100	68.52
	IBuILD	**100**	**78.13**
City Centre	Gálvez-López [49]	100	30.61
	FABMAP 2.0 [31]	100	38.77
	IBuILD	**100**	**38.92**



(a) Detected loop closures on City Centre dataset (b) Detected loop closures on Malaga6L dataset

Fig. 4.18: Loop closures detected (marked in red) by the proposed approach on the map of City Centre and Malaga6L dataset

Results for City Centre and Malaga6L Dataset

Figure 4.17(a) shows the precision and recall of the proposed approach for different δ thresholds on the above mentioned datasets. The maximum possible recall rate with 100% precision is mentioned in Table 4.2. The results mentioned in Table 4.2 (for FABMAP 2.0 and the approach proposed by Gálvez-López) have been taken from [49] as the same script and groundtruth has been used for evaluation. It can be seen that the proposed approach is capable of producing higher recall with 100% precision in comparison to other methods. Figure 4.17(b) shows the evolution of the vocabulary size for the precision and recall highlighted in Table 4.2. Figure 4.18 shows the loop closures detected by the approach in red on the City centre and Malaga6L trajectory. Since Malaga6L is the largest dataset (containing 3474 images) used in this paper, the execution time of the entire pipeline is mentioned in milliseconds in Table 4.3. The computation time of the entire pipeline is around 50 millisecond on average per image. Figure 4.19 shows example images of the loop closures detected by the proposed approach on the City Centre and Malaga6L dataset.

4.6.2 Place Recognition using Active and Passive sensors

The objective of the experimental evaluation in this subsection is to highlight the importance of laser intensities for place recognition algorithms in comparison to other forms of input data such as camera images or depth information from laser scanners. The evaluation

Tab. 4.3: Average execution time (Milli Sec) for a single image on Malaga6L dataset

Property	Keypoint detection	Descriptor extraction	Clustering	Assignment to BOW	Loop closure hypothesis + Evaluation	Vocabulary update
Mean	3.4	1.9	0.038	45	0.1120	0.0088
Standard deviation	0.45	0.44	0.068	37	1.5	0.063

is carried out using different formats of input data such as images or point clouds given different projection models and scene descriptions under challenging lighting conditions. To evaluate the place recognition pipeline a point cloud dataset is acquired in a stop-scan-go manner near TUM campus in Munich due to lack of publicly available datasets with laser intensities in outdoor urban environments. The first part of the dataset was acquired during day time at different locations near the TUM (Technische Universität München)

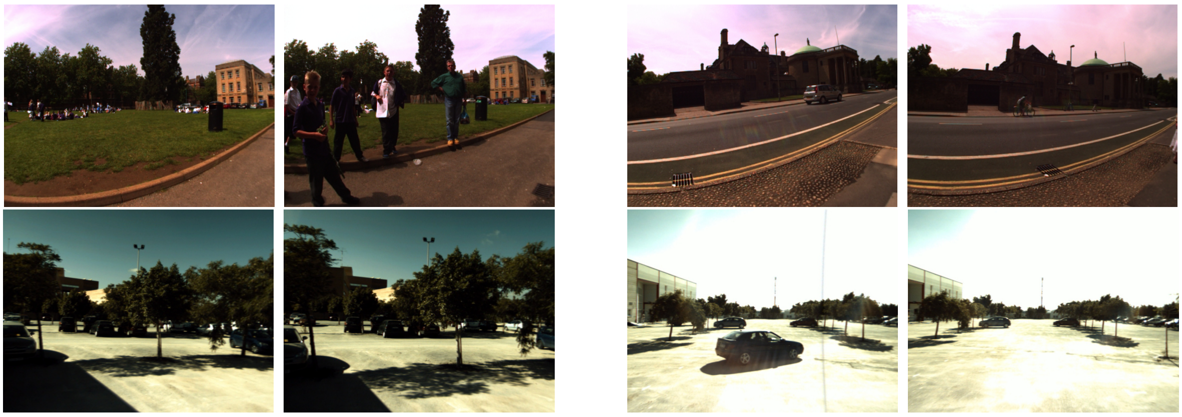


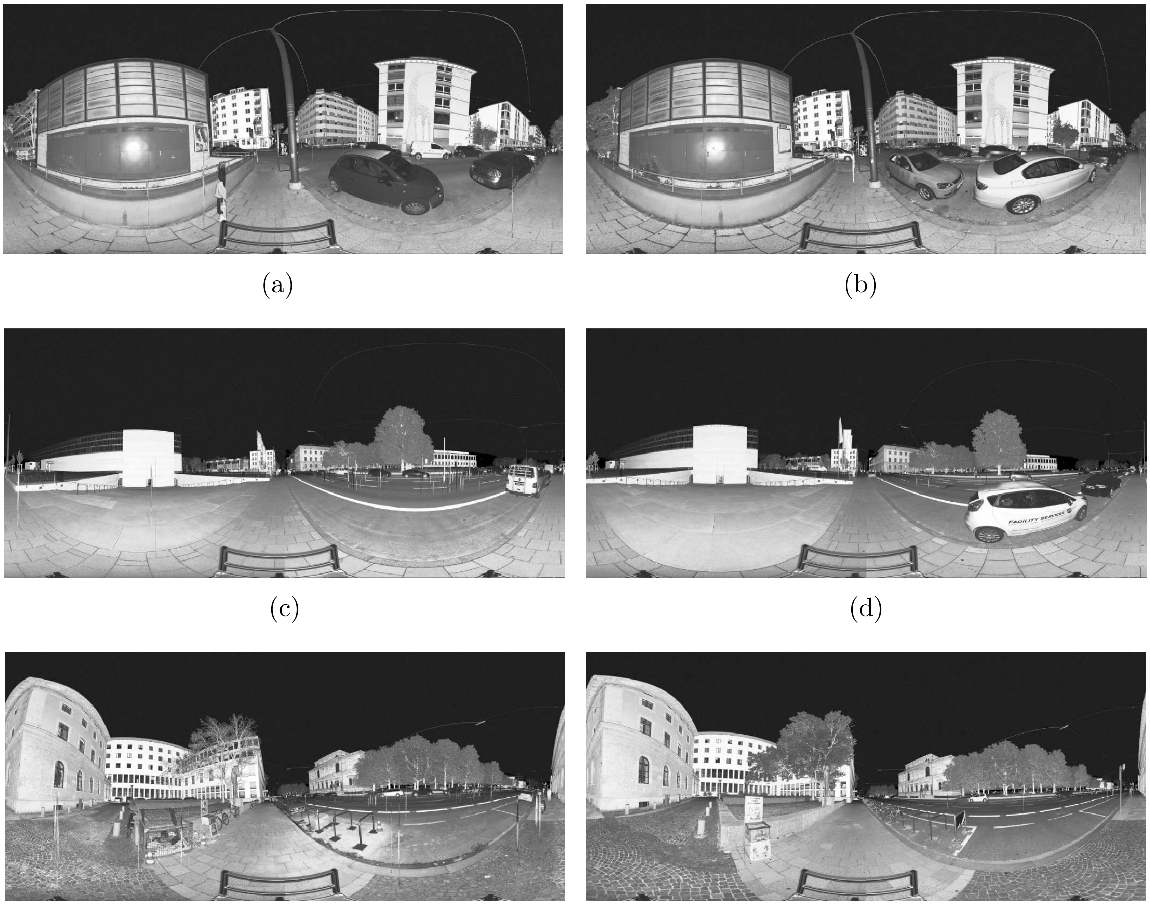
Fig. 4.19: An example of loop closure detected by the proposed approach on the City Centre and Malaga6L dataset.

campus using the Z+F 5010C scanner. In the second run those locations were revisited during the night time after period of 3 months. The dataset consists of 86 scans covering an overall area of 0.3 km^2 with each scan containing on average 20 Million points over a range of 150 meters.

The point clouds acquired in the dataset do not follow a temporal sequence (to evaluate the performance of the algorithm in context of the kidnapped robot problem), hence the robot position *jumps* to different locations near the TUM campus in Munich. Figure 4.20 shows different instances of the places visited around the TUM campus. To perform a comparison with *passive* sensors (cameras), images were also acquired at the same locations under the same lighting conditions with a Canon EOS 5D camera with a long exposure time as shown in Figure 4.21.

The precision-recall of the place recognition algorithm is determined based on the groundtruth, which is generated by manually inspecting the point clouds and considering them to be the same places if the distance between the locations is *less* than 5 meters. To evaluate and compare the precision-recall curves, two main criterion have been chosen, specifically the *maximum achievable* recall/precision generated by the algorithm at 100% precision/recall respectively. The above mentioned regions of the precision-recall curve are critical for place recognition algorithms because higher recall at 100% precision means that the algorithm is capable of generating a larger number of true positives without any false positives. In contrast high precision at 100% recall means that the algorithm is capable of recognizing all similar places albeit with a certain amount of false positives.

This subsection is further divided into three main parts. The first part focuses on highlighting the importance of intensities by comparing the performance of the place recognition algorithms (based on local or global descriptors described) using intensity images in comparison to range images acquired from laser scanners. The second part compares the performance of the place recognition approaches using intensity images in contrast to camera images (passive sensor) whereas the final part highlights the importance of using laser intensities for point cloud based place recognition using the BOW approach. All three parts of this subsection highlight important aspects/properties of the scene description and



(a) (b)

(c) (d)

(e) (f)

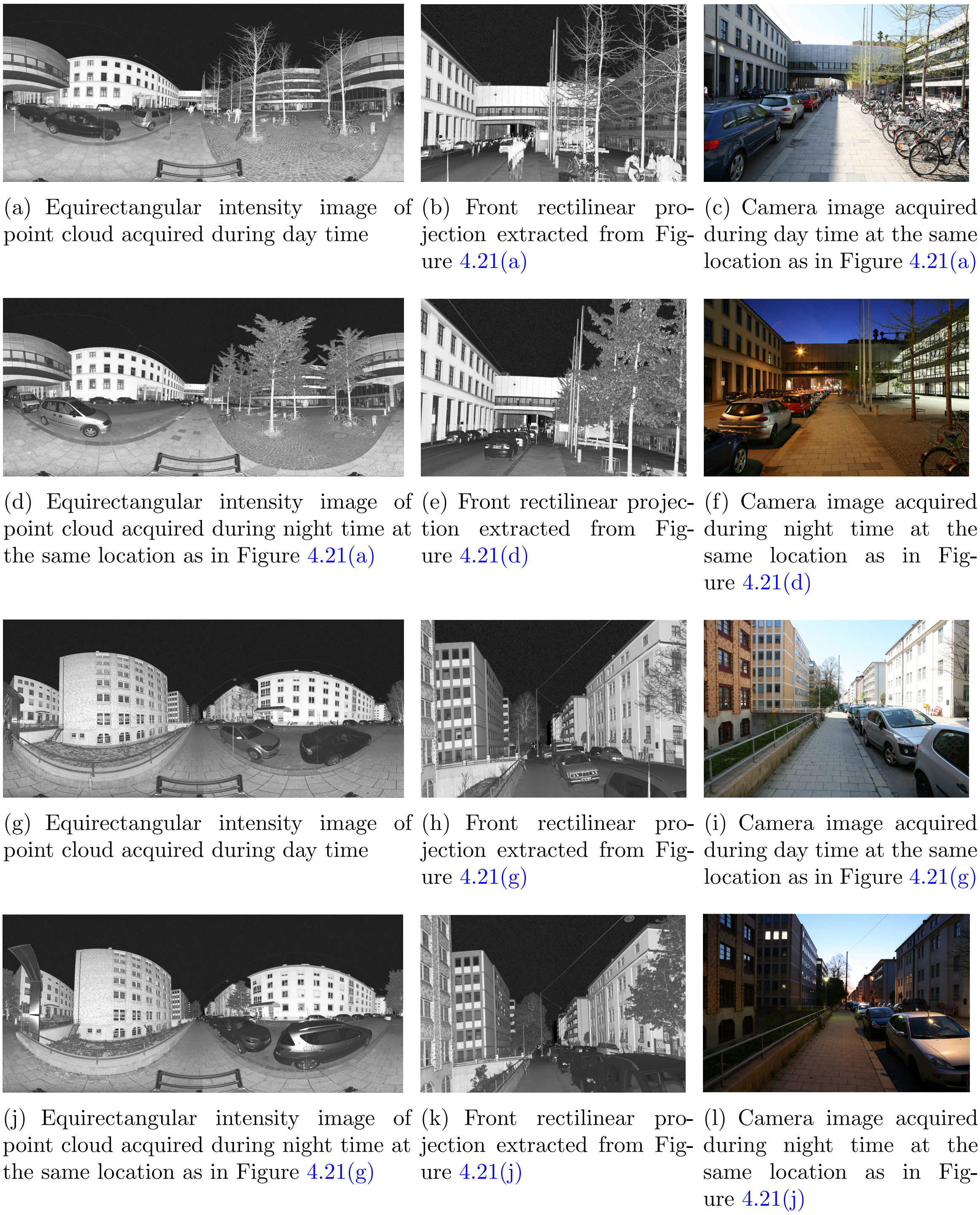
Fig. 4.20: a,c,e) Equirectangular intensity images of point clouds acquired during the day time. b,d,f) Intensity images of point clouds acquired during the night time. The long term dynamics such as parked cars moving away are visible in the images.

projection models in context of place recognition. It is important to specify that the same pipeline and parameter values are used to compare the precision-recall curves of different place recognition approaches (based on local or global descriptors) when intensity/range images from laser scanners or camera images are used as input (unless otherwise specified).

Comparison of Place Recognition Approaches using Equirectangular Range and Intensity Images

This part of the subsection evaluates the performance of place recognition pipeline based on local and global descriptors (described in Section 4.5) when equirectangular intensity and range images are used as input. To perform a fair comparison, the precision-recall curves of the place recognition algorithms are compared for the same parameters values.

- **Local Descriptors (BOW approach)**



(a) Equirectangular intensity image of point cloud acquired during day time

(b) Front rectilinear projection extracted from Figure 4.21(a)

(c) Camera image acquired during day time at the same location as in Figure 4.21(a)

(d) Equirectangular intensity image of point cloud acquired during night time at the same location as in Figure 4.21(a)

(e) Front rectilinear projection extracted from Figure 4.21(d)

(f) Camera image acquired during night time at the same location as in Figure 4.21(d)

(g) Equirectangular intensity image of point cloud acquired during day time

(h) Front rectilinear projection extracted from Figure 4.21(g)

(i) Camera image acquired during day time at the same location as in Figure 4.21(g)

(j) Equirectangular intensity image of point cloud acquired during night time at the same location as in Figure 4.21(g)

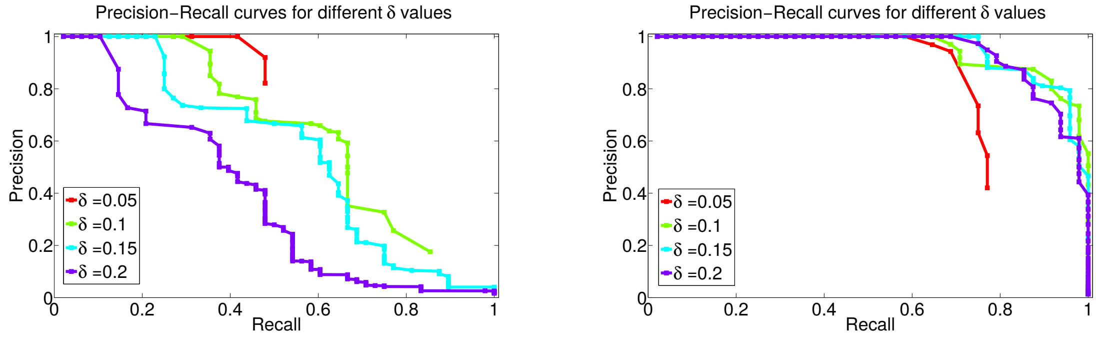
(k) Front rectilinear projection extracted from Figure 4.21(j)

(l) Camera image acquired during night time at the same location as in Figure 4.21(j)

Fig. 4.21: a-d), g-j) Equirectangular intensity image of the same location during day and night time. b-e), h-k) Front rectilinear projection extracted from a-d, g-j respectively. c-f), i-l) Camera images at the same location as in a-d, g-j under similar lighting conditions.

The evaluation performed in this part of the subsection compares the performance of the BOW place recognition approach defined in Section 4.5.2 using range (${}^{\mathrm{r}}\mathbf{I}^{\mathrm{eqrect}}$) and intensity

(${}^{\mathrm{i}}\mathbf{I}^{\mathrm{eqrect}}$) images as input. Figure 4.22 shows the precision-recall curves for different δ values for the BOW place recognition approach. It can be seen that given any δ threshold, the BOW place recognition algorithm performs better using intensity images as input in comparison to range images. The intensity based BOW approach is capable of generating higher recall at 100% precision and vice versa than its counterpart. The main reason for the improvement in the performance of the place recognition algorithm is due to the capacity of laser intensities as an intrinsic surface property to differentiate or find similarity between different environment scenes. In contrast, the range information can be ambiguous for local descriptors as many scenes in the Munich campus dataset have a similar structure.



(a) BOW using range images

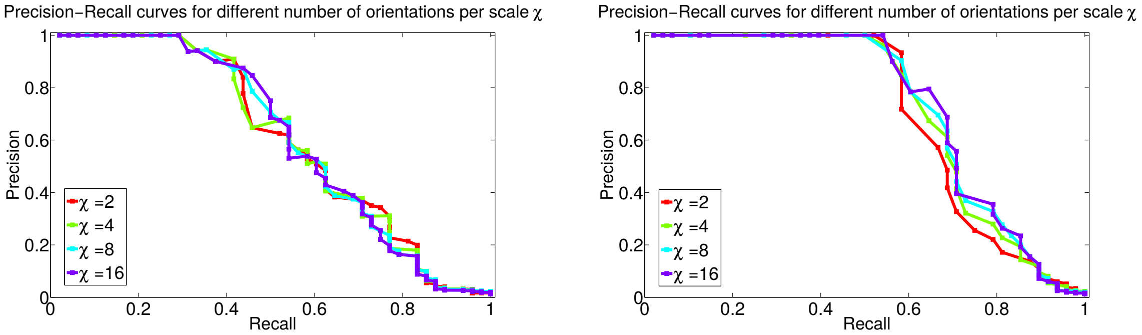
(b) BOW using intensity images

Fig. 4.22: a) The precision-recall curve of the BOW approach using range (${}^{\mathrm{r}}\mathbf{I}^{\mathrm{eqrect}}$) images for different δ values. b) The precision-recall curve of the BOW approach using intensity (${}^{\mathrm{i}}\mathbf{I}^{\mathrm{eqrect}}$) images for the same parameters as shown in Figure 4.22(a). It can be seen that using intensity images as input improves the precision-recall of the place recognition algorithm significantly.

- **Global Descriptors (HOG and GIST):**

Figure 4.23 and Figure 4.24 show the precision-recall curves for the GIST and HOG based place recognition approach using intensity and range images as input for the same parameter values. It is important to highlight here that the evaluation for GIST and HOG (using range and intensity images) was carried out for a large set of parameter values of which only a subset is being shown here. In all evaluations it is found that using intensity images as input improves the performance of the place recognition approach. It can be seen in the above mentioned figures that for all set of parameters (χ for GIST and L for HOG) the place recognition algorithm using intensity images as input generates higher recall at 100% precision in comparison to range images.

An important aspect to highlight here is that the global descriptor based place recognition algorithms (GIST and HOG) never achieve high precision at 100% recall in comparison to the evaluation of the BOW approach shown in Figure 4.22. The reason due to which this happens is because the Munich campus dataset contains scans of the same location in which the observer orientation is changed by 180° as shown in Figure 4.25(a) and Figure 4.25(b). The *global/holistic* descriptors summarize an image by generating a global



(a) GIST based place recognition using range images

(b) GIST based place recognition using intensity images

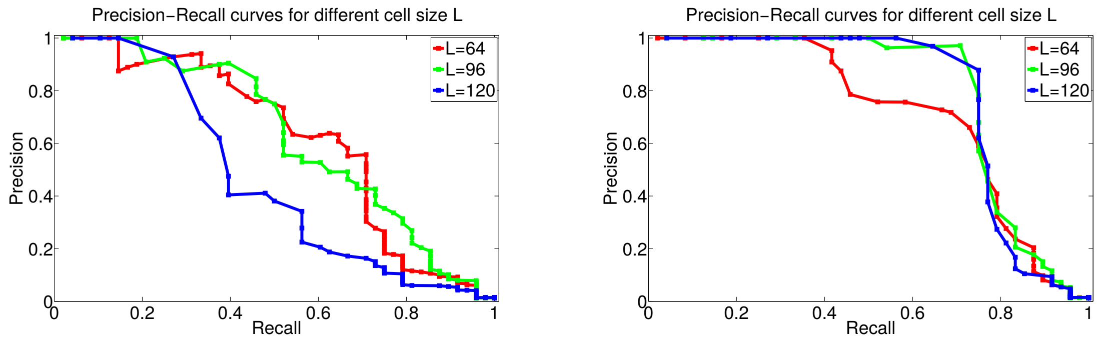
Fig. 4.23: a) The precision-recall curve of the GIST based place recognition approach using range (${}^{r}\mathbf{I}^{\text{eqrect}}$) images given different orientations per scale χ with a fixed number of blocks and scales i.e. φ and Ω respectively. b) The precision-recall curve of the GIST based place recognition approach using intensity (${}^{i}\mathbf{I}^{\text{eqrect}}$) images as input for the same set of parameters as shown in Figure 4.23(a). It can be seen that using intensity images improves the maximum achievable recall of the place recognition approach at 100% precision. It is important to highlight that the above mentioned evaluation was performed for a large number of parameter values (χ, φ and Ω) of which a very small subset is shown here for conciseness. In all evaluations it is found that using intensity images as input to the GIST based place recognition approach improves performance in comparison to range images.

representation which is processed in a predefined order (such as accumulating histograms for each cell in an image and appending them to form a vector in case of HOG) which makes them sensitive to the observer orientation even though equirectangular projections provide a 360° field of view.

Comparison of Place Recognition Approaches using Camera Images and Laser Scanner Intensity Images

This part of the subsection compares the performance of the place recognition algorithms based on *local* and *global* descriptors using laser scanner intensity and camera images as input. It is important to specify that the color images from the camera were passed through the same pipeline as shown in Figure 4.3 with the data preprocessing block only converting the color image to an 8 bit grayscale image (the intensity image is also a 8 bit grayscale image). The equirectangular projection of the point cloud provides a panoramic perspective whereas cameras have a limited field of view, hence a rectilinear projection is calculated to perform a fair comparison. Figure 4.25 shows the equirectangular, rectilinear projection as well as the camera images acquired at the same location.

As mentioned earlier, the Munich dataset contains scans in which the observer orientation was varied by 180° as shown in Figure 4.25(a) and 4.25(b). Unlike the equirectangular projection which has a wide field of view (panorama), the camera has a limited field of view which can cause the place recognition algorithm to not recognize the same location



(a) HOG based place recognition using range images

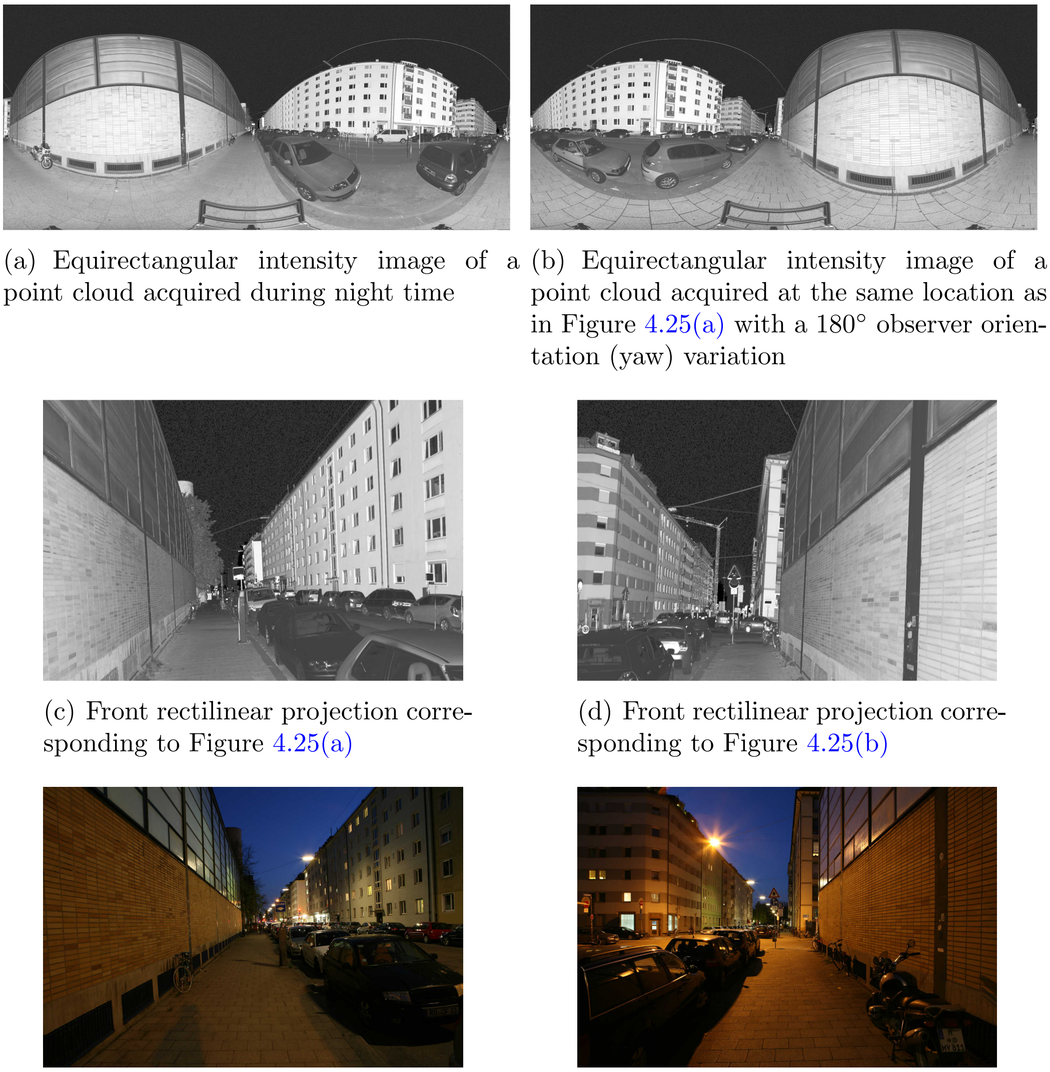
(b) HOG based place recognition using intensity images

Fig. 4.24: a) The precision-recall curve of the HOG based place recognition approach using range (${}^{\mathrm{r}}\mathbf{I}^{\mathrm{eqrect}}$) images for different cell size L with a fixed size of spatial support σ for the Gaussian filter. b) The precision-recall curve of the HOG based place recognition approach using intensity (${}^{\mathrm{i}}\mathbf{I}^{\mathrm{eqrect}}$) images for the same set of parameters as shown in Figure 4.24(a). It can be seen that using intensity images improves the maximum achievable recall of the place recognition approach at 100% precision. It is important to highlight that the above mentioned evaluation was performed for a large number of parameter values (L, and σ) of which a very small subset is shown here for conciseness. In all evaluations it is found that using intensity images as input to the HOG based place recognition approach improves performs in comparison to range images.

as discussed in Figure 4.25. Hence, the experimental evaluation is carried out for two different cases, firstly when the place recognition algorithm is provided only with the front rectilinear image (with a limited field of view). In contrast, the second case considers the scenario in which the place recognition algorithm is provided with the front and back (corresponding to 180° observer yaw variation) rectilinear projection or camera images. A true positive occurred in this evaluation if images from any of the observer orientations was correctly recognized. The comparative evaluation of both cases mentioned above is only shown for local descriptor based place recognition approach (for conciseness), however the conclusion is valid for global descriptors as well.

- **Local Descriptors (BOW approach):**

In this part of the subsection, the performance of the BOW place recognition algorithm is compared when camera (${}^{\mathrm{c}}\mathbf{I}$) and intensity (${}^{\mathrm{i}}\mathbf{I}^{\mathrm{rect}}$) images are used as input. Figure 4.26 shows the precision-recall curves of the BOW approach for different δ thresholds using camera and intensity images as input for the two different cases (a-b and c-d) mentioned above. Firstly it can be seen that formulating the BOW place recognition algorithm over intensity images significantly improves the precision and recall of the algorithm for both cases (Figure 4.26(b) in comparison to Figure 4.26(a) and Figure 4.26(d) in comparison to Figure 4.26(c)). Secondly, it can be seen from the figures that the maximum achievable recall by the algorithm at 100% precision increases in the second case (when front and back



(a) Equirectangular intensity image of a point cloud acquired during night time

(b) Equirectangular intensity image of a point cloud acquired at the same location as in Figure 4.25(a) with a 180° observer orientation (yaw) variation

(c) Front rectilinear projection corresponding to Figure 4.25(a)

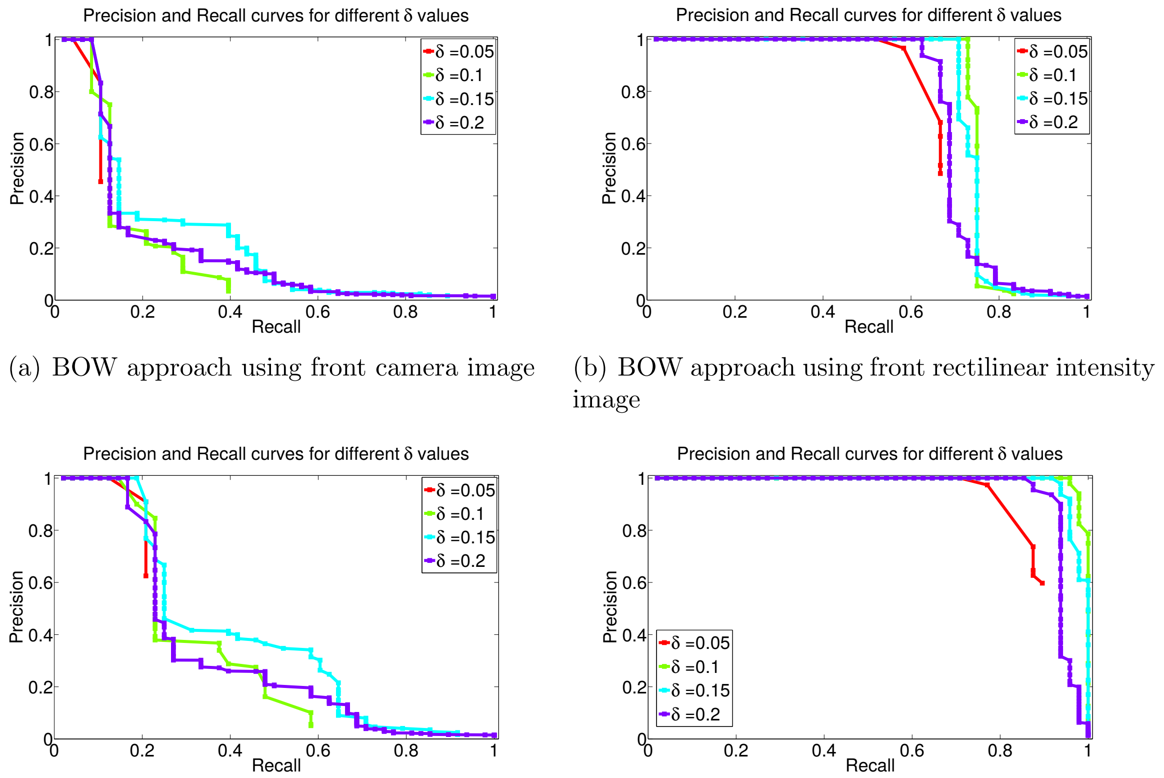
(d) Front rectilinear projection corresponding to Figure 4.25(b)

(e) Camera image at the same location as shown in Figure 4.25(a)

(f) Camera image at the same location (with the same orientation) as in Figure 4.25(b)

Fig. 4.25: a-b) The Munich campus dataset contains scans in which the observer orientation (yaw) is changed by 180° at the same location. Due to the limited field of view of the rectilinear projection (c-d) and the camera (e-f), place recognition approaches are unable to determine if the observer is at the same location.

rectilinear images are used) in comparison to the first case. It can be seen in Figure 4.26(b) that the BOW approach using the front rectilinear intensity image does not generate high precision at 100% recall due to the issue highlighted in Figure 4.25, however when the place recognition algorithm is provided with the front and back rectilinear intensity images the precision and recall of the algorithm improves by a significant margin as shown in Figure 4.26(d). The BOW place recognition approach using camera images does not perform well for the same parameters due to lack of feature repeatability under drastic illumination conditions [48, 116].



(a) BOW approach using front camera image

(b) BOW approach using front rectilinear intensity image

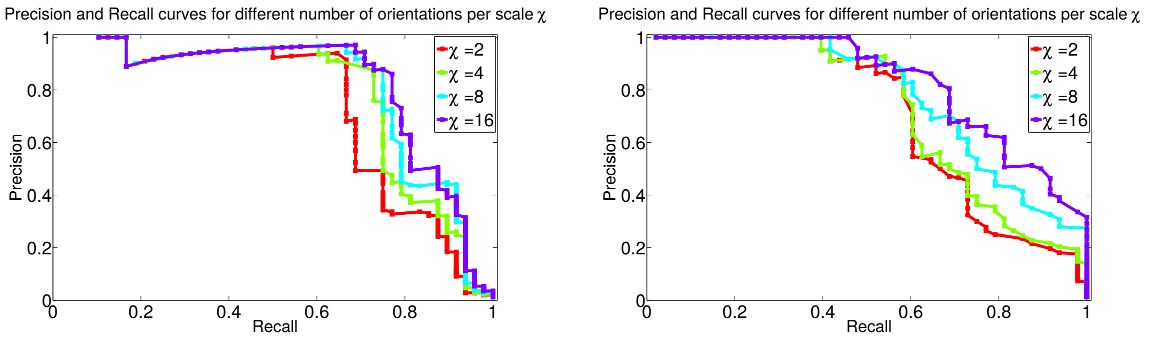
(c) BOW approach using front and back camera images

(d) BOW approach using front and back rectilinear intensity images

Fig. 4.26: a-b) The performance of the BOW place recognition algorithm for the first case when it uses only the front rectilinear projection or the front camera image c-d) The second case when it uses both the front and back rectilinear projections or camera images. It can been seen that the place recognition algorithm using intensity images outperforms the same approach using camera images (compare Figure 4.26 b) with a) and d) with c)).

- **Global Descriptors (GIST and HOG):**

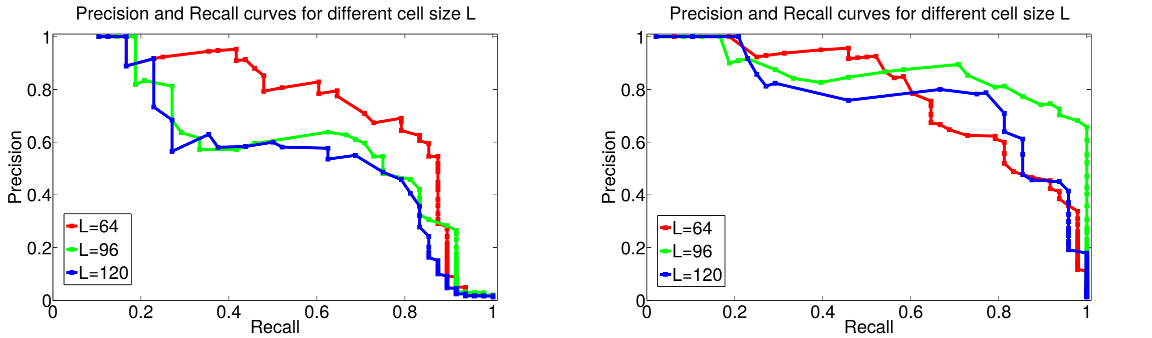
Figure 4.27 and Figure 4.28 shows the precision-recall of the place recognition approach based on GIST and HOG global descriptors respectively using camera (${}^{c}\mathbf{I}$) and intensity (${}^{i}\mathbf{I}^{rect}$) images as input. Both figures show that the place recognition approach using intensity images is capable of generating higher precision at 100% recall and vice versa in comparison to the approach that uses camera images as input for the same parameter values. It can be seen that as the place recognition approach based on global descriptors is provided with front and back intensity images (increasing the field of view and resolving the orientation issue highlighted in Figure 4.25) they are capable of generating high precision at 100% recall in contrast to the evaluation in Figure 4.23 and Figure 4.24. It is important to highlight that for any set of parameters the place recognition pipeline using camera images does not generate 100% recall at high precision. Although in context of GIST descriptor based place recognition using camera images the algorithm generates higher precision and recall in a small region of the curve in comparison to the approach based



(a) GIST based place recognition using front and back camera images

(b) GIST based place recognition using front and back rectilinear intensity images

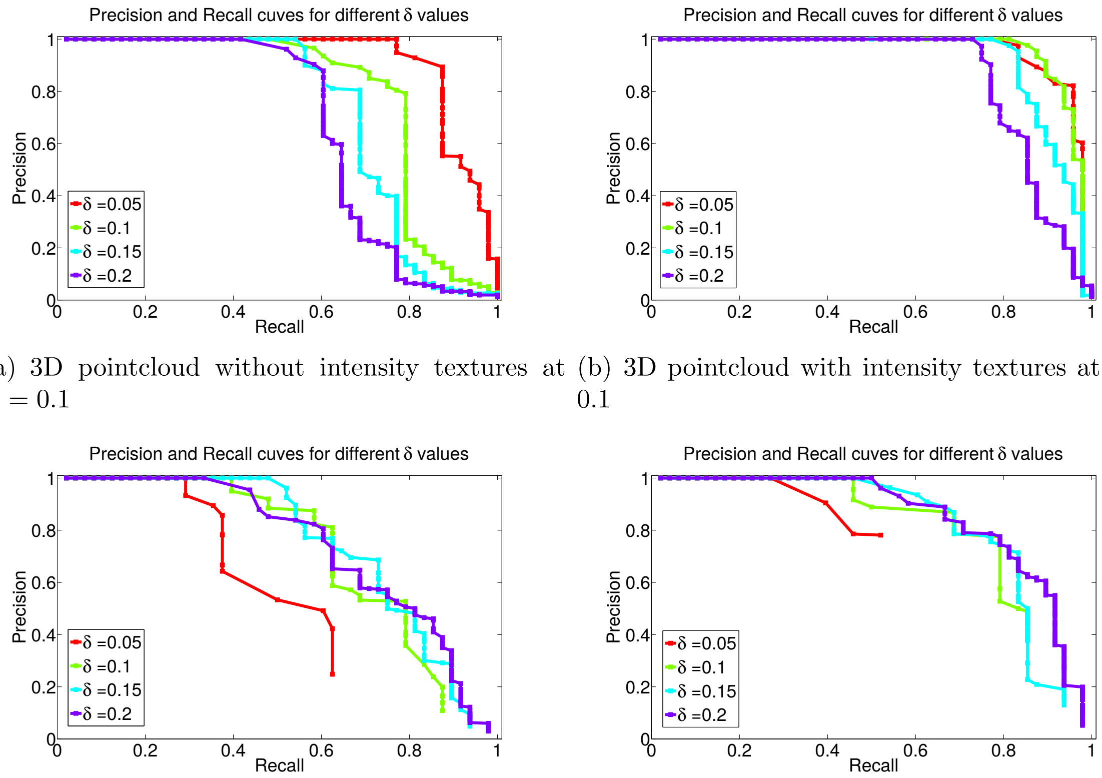
Fig. 4.27: a-b) GIST based place recognition using camera (${}^{c}\mathbf{I}$) and intensity (${}^{i}\mathbf{I}^{\text{rect}}$) images respectively for different number of orientations per scale χ with other parameters being the same. It can be seen that place recognition approach using intensity images is capable of generating higher recall at 100% precision and vice versa in comparison to its counterpart using camera images. It is important to highlight that the above mentioned evaluation was performed for a large number of parameter values (χ, φ and Ω) of which a very small subset is shown here for conciseness. In all evaluations it is found that using intensity images as input to the GIST based place recognition approach improves performance in comparison to camera images.



(a) HOG based place recognition using front and back camera images

(b) HOG based place recognition using front and back rectilinear intensity images

Fig. 4.28: a-b) HOG based place recognition using camera (${}^{c}\mathbf{I}$) and intensity (${}^{i}\mathbf{I}^{\text{rect}}$) images for different cell size L with other parameters being the same. It can be seen that the place recognition using intensity images is capable of generating higher precision at 100% recall in comparison to its counterpart using camera images. It is important to highlight that the above mentioned evaluation was performed for a large number of parameter values (L, and σ) of which a very small subset is shown here for conciseness. In all evaluations it is found that using intensity images as input to the HOG based place recognition approach improves performs in comparison to camera images.



(a) 3D pointcloud without intensity textures at $\psi = 0.1$

(b) 3D pointcloud with intensity textures at $\psi = 0.1$

(c) 3D pointcloud without intensity textures at $\psi = 0.5$

(d) 3D pointcloud with intensity textures at $\psi = 0.5$

Fig. 4.29: The performance of the BOW place recognition approach based on 3D point clouds using SHOT descriptors with and without intensity textures. It can be seen that for lower ψ (the grid cell size used to downsample the pointcloud) values ($\psi = 0.1$ m) the BOW place recognition algorithm using intensity textures is capable of generating higher recall at 100% precision and vice versa. b-d) The decrease in performance of the BOW place recognition algorithm is due to loss in information about the environment geometry (due to downsampling).c-d) The precision-recall performance gap of the BOW place recognition approach with and without intensity textures also decreases due to loss in information about the environment geometry.

on intensity images, however at the most important parts of the precision recall curve (maximum achievable recall at 100% precision or maximum achievable precision at 100% recall) the intensity based place recognition performs better than its counterpart.

- **Comparison of the Point Cloud based Place Recognition Approach with and without Intensities**

This part of the subsection focuses on evaluating the BOW place recognition approach using descriptors extracted from the actual point cloud without using any projection. The BOW approach uses SHOT descriptors as they have been shown to have higher repeatability and discriminative capabilities in comparison to other descriptors [164].

The comparison performed in this subsection evaluates the performance of the 3D point cloud based BOW place recognition approach with and without (only geometry) intensity textures [164]. The process of descriptor extraction includes keypoint extraction from the point cloud and furthermore calculation of the SHOT descriptor for each keypoint using the neighbourhood defined by a radius of $5 \times \psi$. Figure 4.29 shows the precision and recall curves of the BOW approach using 3D descriptors for different downsampled pointclouds (with and without intensity textures) using grid cell size ψ. Figure 4.29 highlights different important aspects, firstly the performance of the place recognition algorithm based on intensity textured point cloud is better (higher recall at 100% precision and vice versa) than its counterpart (point cloud without texture) for $\psi = 0.1$ m. Secondly, it can be seen that as the ψ value is increased the performance of the place recognition algorithm decreases significantly due to loss in information about the environment geometry (due to downsampling). In addition, it can also be seen that due to this downsampling the precision-recall performance gap of the BOW place recognition approach with and without intensity textures also decreases.

4.7 Discussion

This section highlights the different characteristics of the proposed binary vocabulary generation mechanism for loop closure detection as well as the proposed generic place recognition pipeline.

4.7.1 Binary Bag of Words Vocabulary Generation for Loop Closure Detection

The experimental evaluation of Section 4.6.1 raises two important issues about the proposed approach: Firstly, the issue of scalability, i.e. handling large vocabularies and secondly the selection of an appropriate δ threshold.

Scalability

The scalability issue can be addressed by formulating an incremental version of the 'vocabulary tree' [141] suitable for binary descriptors. The advantage of such an adaptation would be to reduce the computational complexity (reducing it to logarithmic instead of linear complexity) during the BOW assignment process discussed in Section 4.4.2 and allow the approach to scale well for large scale datasets and vocabularies containing 1 million or more words.

Distance threshold

Consider the second issue of selecting an appropriate δ (distance) threshold. The factors that influence the δ threshold include the operating conditions i.e. lighting conditions as current state of the art feature detectors are not completely invariant to such changes and the amount of overlap present between images for feature tracking. In principle, a simple mechanism can be used to estimate the δ threshold for a particular dataset. This mechanism requires matching descriptors (using a specific δ threshold) between a pair of consecutive images and reducing the δ threshold until the false matches are eliminated. It is important that this pair should be a true representative of the operating conditions and expected overlap between images in that dataset.

4.7.2 Place Recognition using Passive and Active Sensors

The experimental evaluation in Section 4.6.2 highlights different aspect of the generic place recognition pipeline which are discussed in detail in this subsection.

Active vs Passive sensors

The decision to formulate the place recognition problem using an active sensor (specifically laser intensities) is due to its invariance to ambient lighting conditions and its dependence on an intrinsic environment property (surface reflectivity). *The advantage as shown in Section 4.6.2 is that it is possible to use a visual vocabulary (based on local or global descriptors) generated during day time to recognize the same places during night time without any preprocessing.* Hence, a visual vocabulary generated using laser intensities is compact and allows better generalizability as it encodes the same location using similar features under varying lighting conditions due to its invariance property. In contrast place recognition approaches based on passive sensors require specific pre-processing or training data under different environment appearances to handle such scenarios. In effect a visual vocabulary generated from passive sensors using training data under different lighting conditions learns to encode the same location with a diverse set of features (thereby generating a large vocabulary). In addition, their exists no notion on the quantity and diversity (under different lighting conditions) of training data that would be sufficient for the vocabulary generation process (in context of passive sensors) to operate under all possible lighting conditions. The above mentioned issue of feature repeatability and matching under adverse lighting conditions in context of passive sensors has been discussed in literature [48, 116, 135].

Offline vs Online Vocabulary Generation

The decision of generating a visual vocabulary in an online, incremental manner (in context of active sensors and the proposed approach) is due to its suitability for online robotic and computer vision applications such as place recognition within SLAM or SFM (structure from motion). An advantage of online, incremental vocabulary generation process is that it removes the inconvenience of collecting a large training dataset for offline processing. The basic idea behind the ideal characteristics discussed in Section 4.1 is that it is *desirable* that a place recognition algorithm should have the capacity to function properly in case a training dataset might not be available for vocabulary generation. *In contrast if training data is available a priori, it is always possible to leverage offline processing to generate a visual vocabulary (using standard mechanisms such as Kmeans) and furthermore adapt it in an online, incremental manner using the mechanism defined in Section 4.5.2.*

Image (Projection) vs 3D Point Cloud based Place Recognition

The proposed pipeline uses different formats of input data such as images (visual appearance) or 3D point clouds (geometry with or without texture) for place recognition.

The main advantage of generating a projection of the point cloud is that it reduces the dimensionality (3D to 2D) of the problem. In addition, it allows the ease of working with commonly used image processing and feature extraction techniques which have been researched and tested extensively by the computer vision and robotics community. In principle, the specific projection (equirectangular or rectilinear) being used also plays a critical role in defining the field of view available to the place recognition algorithm as discussed in Section 4.6.2. Generating a projection has its disadvantages as well because of the variation in appearance due to changes in observer position/orientation. This variation in appearance can be problematic for place recognition algorithms as local descriptors are shift invariant to a certain degree whereas the performance of global descriptors degrades significantly with view point changes. The main advantage of 3D point cloud descriptors is their invariance (to a large extent) to the observer pose variation (translation and rotation). In addition, the formulation of the place recognition problem over 3D point clouds simplifies the estimation of the relative transform between the recognized places (in contrast the projection leads to the loss of information about the environment geometry). Given the descriptor correspondences and the keypoint locations (where the descriptors were extracted), the relative transform between two point clouds can be extracted using a closed form solution [104]. The limiting factor in the performance of 3D descriptors is their sensitivity to the point cloud density as shown in Section 4.6.2 as well as noise. In principle the decision to formulate the place recognition problem over images or point clouds is a design choice that is highly dependent on the desired characteristics of the place recognition algorithm.

Applicability of the Temporal Consistency Constraint over Sensor Observations, Odometry and GPS

The objective of removing the temporal consistency contraint over sensor observations, odometry and GPS is to highlight the discriminative abilities of intensities and its reliability, robustness for global place recognition. Another perspective of viewing the above mentioned aspect is to consider an generic application (outside the scope of robotics) in which point clouds/images similar to a given target point cloud are retrieved from a database based on a similarity metric.

In a typical robotics scenario it is always advisable to fuse information from multiple sources to increase robustness, hence in context of real application the proposed pipeline should always be used in conjunction with additional sensors (such as GPS, temporal consistency as well as odometry). The incorporation of the temporal consistency constraint within the proposed pipeline is quite simple. *The temporal consistency constraint as proposed in [118] can be applied by introducing a constant velocity model which limits the search space of the place recognition hypotheses to a line in the symmetric similarity matrix. In principle, the enforcement of the temporal consistency constraint makes the place recognition problem simpler as it limits the search space for the next candidate in the place* *recognition hypotheses.*

4.8 Conclusion

This section presents the main conclusions of the proposed binary vocabulary generation mechanism as well as highlights the advantages and characteristics of the place recognition pipeline based on laser intensities.

4.8.1 Binary Bag of Words Vocabulary Generation for Loop Closure Detection

A subsection of this chapter focused on an online, incremental approach of binary visual vocabulary generation for loop closure detection. The main purpose of focusing on binary descriptor based vocabularies is because they require reduced computational and memory complexity in comparison to real valued descriptors. The proposed binary vocabulary generation process is based on tracking features across consecutive images making it invariant to the robot pose and ideal for detecting loop closures. In addition, a simple mechanism for generating and updating the binary vocabulary is presented which is coupled with a similarity function and temporal consistency constraints to generate loop closure candidates. The proposed approach is evaluated on different publicly available datasets and it has been shown that in comparison to the state of the art it is capable of generating higher recall at 100% precision.

4.8.2 Place Recognition using Active and Passive Sensors

In addition to the loop closure detection pipeline with temporal constraints, this chapter also addresses the problem of place recognition under challenging lighting conditions using active and passive sensors. A generic pipeline for place recognition is presented which can be adapted for different robotic and computer vision applications depending on the desired set of characteristics i.e. the capability of operating under challenging lighting conditions, requirement of any *prior training data*, *odometery, GPS* or any *temporal consistency contraints* over sensor observations. The proposed place recognition pipeline is evaluated on a dataset collected in the city of Munich near the TUM campus in which different locations are visited during the day and later revisited during the night time. The experimental evaluation shows that using intensity images as input in comparison to types of input data, such as camera or range images, is beneficial for the place recognition algorithms (operating with local or global descriptors) operating under challenging lighting conditions. In addition, it shows that given the same place recognition pipeline (based on *local* or *global* descriptors given the same parameter settings), intensities generate better precision-recall curves in comparison to other types of input data. The results also underline the importance of using intensity textured point clouds for 3D point cloud based place recognition. The evaluation also highlights certain design decisions in context of place recognition algorithms such as the strong dependence of global descriptors on observer orientation, the effect of the limited field of view of the rectilinear projection model as well as the decrease in performance due to downsampling of point clouds. In summary, the proposed pipeline based on laser intensities is capable of generating high precision, recall under adverse lighting conditions on a challenging dataset without any requirement of prior training data, odometry, GPS or any temporal consistency constraints.

5 Summary & Conclusions

This section presents a brief summary, conclusions and possible future research directions in context of the main contributions of this thesis.

5.1 Summary

This thesis contributes in the domain of *perception* within the field of mobile robotics by proposing techniques that allow robots to generate accurate maps of the environment. An accurate map is an essential requirement for a wide variety of tasks such as robotic navigation and exploration. This section provides a summary of the main contributions made by this thesis in the areas of *Environment representation*, *Simultaneous Localization and Mapping (SLAM)* and *Loop closure/place recognition detection* for consistent and accurate environment mapping.

5.1.1 Environment representation

This thesis contributes in the domain of grid based environment representation by proposing an approach which is capable of approximating the environment using a variable resolution grid. The proposed approach extends the standard occupancy grid by adding a fusion process based on occupancy probabilities that couples the surface representation, i.e. occupancy probabilities, with the spatial decomposition of the grid thereby generating variable resolution grid based environment representations. Furthermore, the variable resolution grid is stored in a hierarchy of axis aligned rectangular cuboids that is incrementally generated and adapted based on sensor observations. The main characteristics of the proposed approach are

- *Incremental*: Allows incremental generation of the grid and the hierarchy based on sensor observations
- *Flexible*: Provides the flexibility of selecting the maximum number of children per node
- *Multiresolution grid cells*: Capable of modeling a variable resolution grid

In summary, the main contributions of this thesis in context of *environment representation* are as follow

- An approach capable of modeling the environment using a variable resolution grid
- A simplistic fusion process that couples the surface attribute i.e. occupancy probability with the spatial decomposition leading to variable resolution representations of the environment in an *online, incremental* fashion

- An extensive experimental evaluation highlighting the characteristics of the proposed approach on a publicly available dataset

5.1.2 Laser Intensities for SLAM

This thesis contributes in the domain of SLAM by proposing a simple calibration process that allows the robot to acquire a pose-invariant measure of surface reflectivity. A typical laser scanner measures the distance to an object as well as quantifies the received optical power after reflection from the object which is termed as the *remission or intensity* value. The important aspect about intensities is that it is dependent on an intrinsic surface property as well as extrinsic parameters such as distance and angle of incidence. This thesis presents a simple calibration process that allows modeling of the extrinsic parameters to acquire a pose-invariant measure of surface reflectivity. This surface reflectivity measure is furthermore used to simultaneously estimate the robot pose as well as acquire a reflectivity map, i.e. occupancy grid augmented with surface reflectivity information, of the environment.

In summary, the main contributions of this thesis in the domain of SLAM are

- A simplistic calibration process to model extrinsic parameters for acquiring a pose invariant measure of surface reflectivity for different laser scanners

- An extension of Hector SLAM capable of simultaneously estimating the robot pose as well as generating a reflectivity map of the environment

- An extensive evaluation of the calibration process as well as the Hector SLAM extension

5.1.3 Place recognition/Loop closure detection

This thesis contributes towards two different aspects of the place recognition/loop closure problem. The first aspect is related to an online, incremental binary vocabulary generation mechanism for loop closure detection using passive sensors. The main advantage of generating binary vocabularies are that they are computationally and memory efficient in comparison to vocabularies generated using real valued descriptors. The second aspect focuses on highlighting the advantage of using laser intensities for place recognition under challenging lighting conditions in comparison to other types of input data such as camera images or geometry information from laser scanners. The main advantage of laser intensities is that they are invariant to ambient lighting conditions and depend on an intrinsic surface property i.e. surface reflectivity.

In summary, the main contributions of this thesis in the domain of place recognition/loop closure detection are

- An *online, incremental* approach for binary vocabulary generation for loop closure detection

- To highlight the advantage of using laser intensities for place recognition under challenging lighting conditions in contrast to other types of input data such as camera images or geometry information from laser scanners
- An extensive evaluation of the vocabulary generation mechanism and the characteristics of laser intensities using different descriptors, projection models and similarity functions

5.2 Conclusion & Outlook

This subsection provides an overview of the conclusions as well as the possible future research directions in the domain of *Environment representation*, *SLAM* and *Place recognition/loop closure* detection.

5.2.1 Environment representation

The main advantage of the proposed environment representation is that it is capable of modeling the environment using a variable resolution grid which is stored in a hierarchy of axis aligned rectangular cuboids. The proposed approach is flexible in the sense that it allows the user to define the maximum number of children allowed per node for the hierarchy thereby influencing its characteristics such as insertion, access time as well as the number of grid cells required to represent the environment. The evaluation highlights that the proposed approach in comparison to the state-of-the-art Octomap approach requires less number of grid cells and provides faster access times. In addition, the number of inner nodes required to represent the hierarchy is significantly less, however the proposed approach requires higher insertion times as it incrementally generates the hierarchy based on the sensor observations.

The fusion process proposed in this thesis assumes a static environment. Possible future work includes an extension of the fusion process to operate in dynamic environments. In principle this extension can be carried out by monitoring the occupancy values of the fused grid cells and splitting them if these values fall below the fusion threshold. In addition, another important research direction is to incorporate object level dynamics into the environment representation. In such a scenario a classifier would be used to detect objects in the point cloud and furthermore approximate them as a rectangular cuboid and add them to the variable resolution grid. Incorporation of object level hypothesis into the environment representation is an essential step towards semantic mapping and critical for development of intelligent and autonomous robots.

5.2.2 Laser Intensities for SLAM

In context of laser based SLAM, a simple calibration process for extrinsic parameters is proposed that allows the robot to acquire a measure of surface reflectivity. The importance of the proposed calibration process is shown by comparing it to other models which systematically ignore the influence of extrinsic parameters. The results show that extrinsic parameter calibration is essential to acquire a pose-invariant measure of surface reflectivity.

In addition, this reflectivity measure is used in an extension of Hector SLAM in which a robot simultaneously estimates its own pose as well as acquires a reflectivity map of the environment. The proposed Hector SLAM extension has been shown to accurately estimate the robot pose and can be useful in cases when geometry information is ambiguous. The reflectivity maps generated by the proposed approach can be used in a wide variety of robotic applications such as global localization, navigation as well as exploration.

The proposed Hector SLAM extension relies on a cost function based only on surface reflectivity information, hence it would interesting to consider other cost functions that combine reflectivity and occupancy information and evaluate their performance. In addition, the scenario in which the point density is low can be problematic for normal vector estimation thereby causing problems for extrinsic parameter correction. Hence an interesting future research direction would be to switch cost functions based on the point density observed by the robot.

5.2.3 Place recognition/Loop closure Detection

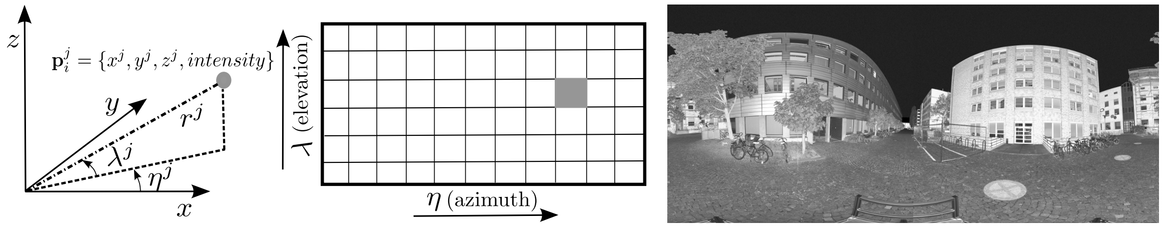
This thesis evaluates and highlights the advantage of laser intensities for place recognition under challenging lighting conditions and compares its performance with other types of input data such as camera images or geometry information from laser scanners. The experimental evaluation shows that using intensity images as input in comparison to other forms of input data, i.e. camera or range images, is beneficial for place recognition algorithms (based on local or global descriptors) operating under challenging lighting conditions. The results also underline the importance of using intensity textured point clouds for 3D point cloud based place recognition. The evaluation highlights certain design decisions in context of place recognition algorithms such as the strong dependence of global descriptors on observer orientation, the effect of the limited field of view of the rectilinear projection model as well as the decrease in performance due to downsampling of point clouds. An interesting future research direction would be to develop approaches that combine the advantage of local and global descriptors for place recognition. It will also be interesting to combine different types of input data such as camera images or intensity information from laser scanners to take advantage of their properties under different conditions.

In context of vocabulary generation mechanisms, the proposed loop closure detection approach shows that it is possible to generate binary vocabularies in an online, incremental manner. The proposed vocabulary generation mechanism coupled with a simple similarity function and temporal consistency constraint is capable of generating high precision-recall on real world datasets in comparison to the state-of-the-art loop closure detection algorithms. A drawback of the proposed vocabulary generation mechanism is the linear complexity in the update process. An interesting research direction would be to develop an approach that allows generation of a binary vocabulary tree in an online, incremental manner thereby reducing the vocabulary update complexity from linear to logarithmic in the number of descriptors present in the vocabulary.

A Image projection models

A.1 Equirectangular projection

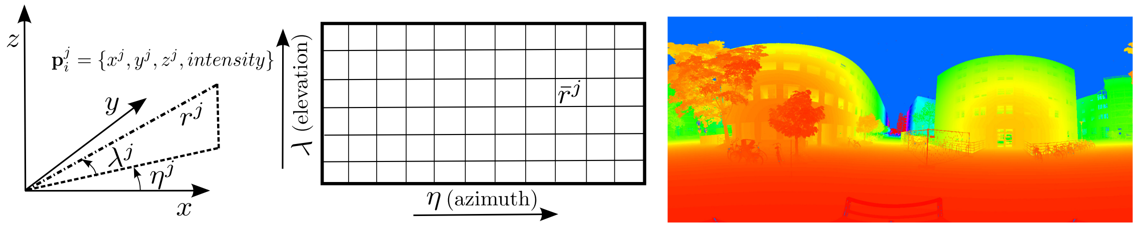
Given the laser scanner observations in cartesian coordinates the first step is to convert them to spherical coordinates defined by range, azimuth and elevation. The intensity, azimuth and elevation of each point observation is used to generate an equirectangular intensity image. It is possible to interpret the azimuth and elevation of the sensor observations as rows and columns of an image respectively and accumulate the intensity value to form a gray scale image as shown in Figure A.1(a). An example of the *panoramic* grayscale image generated via the above mentioned projection is shown in Figure A.1(b) whereas Figure A.4 shows the pseudocode for generating it given a point cloud.



(a)

(b)

Fig. A.1: (a) Laser scanner observation of the j^{th} point in the i^{th} point cloud $\mathbf{P}_i$. (b) Equirectangular intensity image obtained after projecting the point cloud. The azimuth and elevation of the j^{th} point is denoted by η^j and λ^j respectively.



(a)

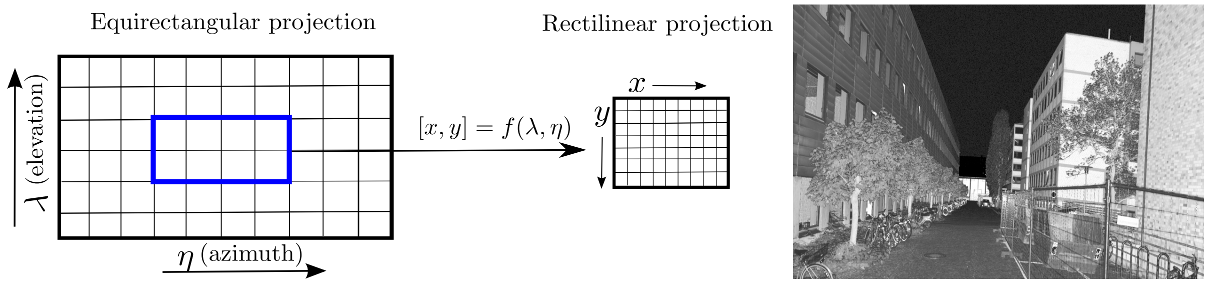
(b)

Fig. A.2: (a) The process of range image generation in which the range value is accumulated in the relevant elevation, azimuth bin. Furthermore, this range image is normalized by the maximum range (as represented by $\bar{r}^j$ in the figure) to generate a matrix of floating point values between 0 and 1. b) (Best visualized in color) An example of the generated range image visualized with a HSV colormap.

Similarly, it is also possible to store the range information at a specific elevation, azimuth bin which leads to the generation of *range images* as shown in Figure A.2(a). The accumulated range value is furthermore normalized by the maximum range generating a matrix of floating point values between 0 and 1. Figure A.2(b) shows the visualization of an equirectangular range image (in a HSV colormap) corresponding to the image shown in Figure A.1(b).

A.2 Rectilinear/Cubic projection

The rectilinear/cubic projection is generated by considering a flat surface tangent to the sphere with the observer viewing from the center. Hence, the main computation involved in this projection is to determine the mapping between the equirectangular coordinates (azimuth and elevation) and the rectilinear image as shown in Figure A.3(a). As the equirectangular projection consists of a 360° panorama, it is possible to generate the rectilinear projection corresponding to different observer orientations (facing forwards or backwards etc. with respect to the principle direction in the equirectangular image). Figure A.3(b) shows the front rectilinear intensity image extracted from the equirectangular image shown in Figure A.1(b) whereas the rectilinear/cubic projection can be extracted for different observer orientations with a predefined horizontal and vertical field of view (fov_h, fov_v in Figure A.4). The cubic projection is a subset of the rectilinear projection in which the horizontal and vertical field of view is set to 90 degrees (given the same image width and height).



(a)

(b)

Fig. A.3: a) An abstract representation of mapping the equirectangular coordinates to the rectilinear image coordinates. b) Front rectilinear projection corresponding to the image shown in Figure A.1(b).

```
equirect_projection($\mathbf{P}_i = \{\mathbf{p}_i^1, \ldots, \mathbf{p}_i^n\}$, width, height)
Input: $\mathbf{P}_i$ // $i^{th}$ point cloud
       width // equirectangular image width
       height // equirectangular image height
Output: $\mathbf{I}_i^{\text{eqrect}}$ // Equirectangular image

Procedure:
//calculate spherical coordinates for all points
$\forall_j \; r^j = \sqrt{p_i^j(x)^2 + p_i^j(y)^2 + p_i^j(z)^2}$
//$p_i^j(x)$: x coordinate of the $j^{th}$ point in the $i^{th}$ point cloud
$\forall_j \; \eta^j = \arctan 2(p_i^j(y), p_i^j(x))$
$\forall_j \; \lambda^j = \arcsin(p_i^j(z)/r^j)$

//calculate azimuth, elevation resolution
res_azimuth $= \dfrac{\max(\boldsymbol{\eta}) - \min(\boldsymbol{\eta})}{\text{width}}$
res_elevation $= \dfrac{\max(\boldsymbol{\lambda}) - \min(\boldsymbol{\lambda})}{\text{height}}$

//convert indices to equirectangular coordinates
$\boldsymbol{\eta} = \dfrac{\boldsymbol{\eta}}{\text{res\_azimuth}}$
$\boldsymbol{\lambda} = \dfrac{\boldsymbol{\lambda}}{\text{res\_elevation}}$

// assign intensity or range to equirectangular coordinate
// $p_i^j$(intensity) : intensity value of the point
$\forall_j \; {}^{\text{intensity}}I_i^{\text{eqrect}}(\eta^j, \lambda^j) \leftarrow p_i^j$(intensity)
        or
$\forall_j \; {}^{\text{r}}I_i^{\text{eqrect}}(\eta^j, \lambda^j) \leftarrow r^j$
return $\mathbf{I}_i^{\text{eqrect}}$;
```

```
rectilinear_projection($\mathbf{I}_i^{\text{eqrect}}$, fov_h, fov_v)
Input: $\mathbf{I}_i^{\text{eqrect}}$ // Equirectangular image
       fov_v // Vertical field of view
       fov_h // Horizontal field of view
Output: $\mathbf{I}_i^{\text{rect}}$ // Front rectilinear image

Procedure:
//elevation ($\lambda$) lies between $[-\frac{\pi}{2}, \frac{\pi}{2}]$
//azimuth ($\eta$) lies between $[-\pi, \pi]$

//calculate upper coordinate ($\mathbf{c}^u$) of rectilinear image
$\mathbf{c}^u = [\tan(\frac{fov_h}{2}), \tan(\frac{fov_v}{2})]$

//generate rectilinear image coordinates
$\mathbf{C} = [\mathbf{c}^1, \ldots, \mathbf{c}^n]$
// $c^1(x) = -c^u(x)$, $c^n(x) = c^u(x)$
// Value of $n$ depends on the size of rectilinear image
// $c^1(y) = -c^u(y)$, $c^n(y) = c^u(y)$
// $\forall_j c^j(z) = 1$

//get equirectangular coordinates from rectilinear image coordinates
$\forall_j \lambda^j = \arctan(c^j(y), \sqrt{c^j(x)^2 + c^j(z)^2})$
$\forall_j \eta^j = \arccos(\frac{-c^j(z)}{\sqrt{c^j(x)^2 + c^j(z)^2}})$

// map values to rectilinear image
$\mathbf{I}_i^{\text{rect}} \leftarrow \mathbf{I}_i^{\text{eqrect}}(\boldsymbol{\eta}, \boldsymbol{\lambda})$
return $\mathbf{I}_i^{\text{rect}}$
```

Fig. A.4: (Left) Pseudocode for generating an equirectangular projection. In case the corresponding elevation (λ), azimuth (η) bin contains multiple observations then a simplistic incremental averaging approach can improve the overall image quality – (Right) Pseudocode for generating a rectilinear projection. The psuedocode shown above calculates the front rectilinear image. The rectilinear images corresponding to different observer orientations can be generated by multiplying the rectilinear image coordinates **C** by the corresponding transformation matrix.

Bibliography

[1] Google self driving car. https://www.google.com/selfdrivingcar/. Accessed: 2016-03-28.

[2] John Amanatides, Andrew Woo, et al. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, volume 87, page 10, 1987.

[3] Adrien Angeli, David Filliat, Stéphane Doncieux, and J-A Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics (TRO)*, 24(5):1027–1037, 2008.

[4] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, 2006.

[5] Andrea Bauer, Klaas Klasing, Georgios Lidoris, Quirin Mühlbauer, Florian Rohrmüller, Stefan Sosnowski, Tingting Xu, Kolja Kühnlenz, Dirk Wollherr, and Martin Buss. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *International Journal of Social Robotics (IJSR)*, 1(2):127–140, 2009.

[6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.

[7] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. *The R*-tree: an efficient and robust access method for points and rectangles*, volume 19. ACM, 1990.

[8] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 5(4):349–359, 1999.

[9] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[10] P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2743–2748, 2003.

[11] Aude Billard and Kerstin Dautenhahn. Experiments in social robotics: grounding and use of communication in autonomous agents. *Adaptive behavior*, 7(LSA3-ARTICLE-2000-003), 2000.

[12] José-Luis Blanco, Francisco-Angel Moreno, and Javier González. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots (AURO)*, 27(4):327–351, November 2009.

[13] Robert Blaskow and Danilo Schneider. Analysis and correction of the dependency between laser scanner intensity values and range. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS)*, 25:107–112, 2014.

[14] Michael Bosse, Paul Newman, John Leonard, and Seth Teller. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research (IJRR)*, 23(12):1113–1139, 2004.

[15] Michael Bosse and Robert Zlot. Keypoint design and evaluation for global localization in 2d lidar maps. In *Robotics Science and Systems (RSS)*, 2008.

[16] Michael Bosse and Robert Zlot. Place recognition using regional point descriptors for 3d mapping. In *Field and Service Robotics (FSR)*, pages 195–204. Springer, 2010.

[17] Frédéric Bourgaul, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 540–545, 2002.

[18] Jack E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[19] Wolfram Burgard, Cyrill Stachniss, Giorgio Grisetti, Bastian Steder, Rainer Kümmerle, Christian Dornhege, Michael Ruhnke, Alexander Kleiner, and Juan D Tardós. A comparison of slam algorithms based on a graph of relations. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2089–2095, 2009.

[20] Martin Buss, Daniel Carton, Sheraz Khan, Barbara Kühnlenz, Kolja Kühnlenz, Christian Landsiedel, Roderick de Nijs, Annemarie Turnwald, and Dirk Wollherr. Iuro–soziale mensch-roboter-interaktion in den straßen von münchen. *at-Automatisierungstechnik*, 63(4):231–242, 2015.

[21] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, pages 778–792. Springer, 2010.

[22] Alexander Carballo, Akihisa Ohya, and Shin'ichi Yuta. People detection using range and intensity data from multi-layered laser range finders. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 5849–5854, 2010.

[23] Daniel Carton, Annemarie Turnwald, Dirk Wollherr, and Martin Buss. Proactively approaching pedestrians with an autonomous mobile robot in urban environments. In *International Symposium on Experimental Robotics (ISER)*, pages 199–214. Springer, 2013.

[24] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[25] Howie M Choset. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[26] C Chow and C Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[27] Peter Corke, Rohan Paul, Winston Churchill, and Paul Newman. Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, pages 2085–2092, 2013.

[28] Cyril Crassin, Fabrice Neyret, Sylvain Lefebvre, and Elmar Eisemann. Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering. In *ACM Symposium on Interactive 3D graphics and games (i3D)*, pages 15–22, 2009.

[29] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, European Conference on Computer Vision (ECCV)*, volume 1, page 22, 2004.

[30] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research (IJRR)*, 27(6):647–665, 2008.

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

[32] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, pages 303–312, 1996.

[33] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.

[34] Kerstin Dautenhahn, M Walters, Sarah Woods, Kheng Lee Koay, Chrystopher L Nehaniv, A Sisbot, Rachid Alami, and Thierry Siméon. How may i serve you?: a robot companion approaching a seated person in a helping context. In *ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 172–179. ACM, 2006.

[35] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1322–1328, 1999.

[36] Frank Dellaert and Michael Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research (IJRR)*, 25(12):1181–1203, 2006.

[37] Albert Diosi and Lindsay Kleeman. Laser scan matching in polar coordinates with application to slam. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3317–3322, 2005.

[38] Ivan Dryanovski, William Morris, and Jizhong Xiao. Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1553–1559, 2010.

[39] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine (RAM)*, 13(99):80, 2006.

[40] Erik Einhorn, C Schroter, and H-M Gross. Finding the adequate resolution for grid mapping-cell sizes locally adapting on-the-fly. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1843–1848, 2011.

[41] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Journal of Computers*, 22(6):46–57, 1989.

[42] Sean P Engelson and Drew V McDermott. Error correction in mobile robot map learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2555–2560, 1992.

[43] Nathaniel Fairfield, George Kantor, and David Wettergreen. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics (JFR)*, 24(1-2):03–21, 2007.

[44] Wei Fang, Xianfeng Huang, Fan Zhang, and Deren Li. Intensity correction of terrestrial laser scanning data by estimating laser transmission function. *International Transactions on GeoSciences and Remote Sensing (TGRS)*, 2014.

[45] Jonathan Fournier, Benoit Ricard, and Denis Laurendeau. Mapping and exploration of complex environments using persistent 3d model. In *IEEE Canadian Conference on Computer and Robot Vision (CRV)*, pages 403–410, 2007.

[46] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *The International Journal of Robotics Research (IJRR)*, 22(12):985–1003, 2003.

[47] K.Ch. Fuerstenberg and K. Dietmayer. Object tracking and classification for multiple active safety and comfort applications using a multilayer laser scanner. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 802–807, June 2004.

[48] Paul Furgale and Timothy D Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics (JFR)*, 27(5):534–560, 2010.

[49] Dorian Galvez-Lopez and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics (TRO)*, 28(5):1188–1197, Oct 2012.

[50] Andrea Garulli, Antonio Giannitrapani, Andrea Rossi, and Antonio Vicino. Mobile robot slam for line-based environment representation. In *IEEE Conference on Decision and Control (CDC)*, pages 2041–2046, 2005.

[51] Biruk Gebre, Hao Men, Kishore Pochiraju, et al. Remotely operated and autonomous mapping system (roams). In *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 173–178, 2009.

[52] Yogesh Girdhar and Gregory Dudek. Online visual vocabularies. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 191–196, 2011.

[53] M Gopi and Shankar Krishnan. A fast and efficient projection-based approach for surface reconstruction. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI) XV*, pages 179–186, 2002.

[54] M Gopi, Shankar Krishnan, and Cláudio T Silva. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Computer Graphics Forum*, volume 19, pages 467–478. Wiley Online Library, 2000.

[55] K Granstrom, Jonas Callmer, Fabio Ramos, and Juan Nieto. Learning to detect loop closure from range data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2009.

[56] Michael Greenspan and Mike Yurick. Approximate kd tree search for efficient icp. In *IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 442–448, 2003.

[57] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine (ITSM)*, 2(4):31–43.

[58] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2432–2437, 2005.

[59] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics (TRO)*, 23(1):34–46, 2007.

[60] Venkat N Gudivada and Vijay V Raghavan. Content based image retrieval systems. *Journal of Computers (JCP)*, 28(9):18–22, 1995.

[61] Jose Guivant, Eduardo Nebot, and Stephan Baiker. Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of robotic systems*, 17(10):565–583.

[62] Jose E Guivant and Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[63] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. *ACM*, 14(2), 1984.

[64] D. Hahnel, W. Burgard, D. Fox, and S. Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 206–211, 2003.

[65] J. Hancock, M. Hebert, and C. Thorpe. Laser intensity-based obstacle detection. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 1541–1546, Oct 1998.

[66] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[67] Daniel M Helmick, Yang Cheng, Daniel S Clouse, Lany H Matthies, Stergios Roumeliotis, et al. Path following using visual odometry for a mars rover in high-slip environments. In *IEEE Aerospace Conference (AeroConf)*, volume 2, pages 772–789, 2004.

[68] M Herbert, C Caillas, Eric Krotkov, In So Kweon, and Takeo Kanade. Terrain mapping for a roving planetary explorer. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 997–1002, 1989.

[69] Kin Leong Ho and Paul Newman. Loop closure detection in slam by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

[70] Kin Leong Ho and Paul Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision (IJCV)*, 74(3):261–286, 2007.

[71] Bernhard Höfle and Norbert Pfeifer. Correction of laser scanning intensity data: Data and model-driven approaches. *ISPRS Journal of Photogrammetry and Remote Sensing (P&RS)*, 62(6):415–433, 2007.

[72] Dirk Holz and Sven Behnke. Sancta simplicitas-on the efficiency and achievable results of slam using icp-based incremental registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1380–1387, 2010.

[73] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots (AURO)*, 34(3):189–206, 2013.

[74] Benjamin Huhle, Martin Magnusson, Wolfgang Straßer, and Achim J Lilienthal. Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4025–4030, 2008.

[75] Albert V Jelalian. *Laser radar systems.* Artech House, 1992.

[76] Edward Johns and Guang-Zhong Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3218, 2013.

[77] Edward Johns and Guang-Zhong Yang. Generative methods for long-term place recognition in dynamic scenes. *International Journal of Computer Vision (IJCV)*, 106(3):297–314, 2014.

[78] Andrew Edie Johnson and Sing Bing Kang. Registration and integration of textured 3d data. *Image and vision computing*, 17(2):135–147, 1999.

[79] Daniek Joubert. *Adaptive occupancy grid mapping with measurement and pose uncertainty.* PhD thesis, Stellenbosch University, 2012.

[80] Simon J Julier and Jeffrey K Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 4238–4243, 2001.

[81] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research (IJRR)*, page 0278364911430419, 2011.

[82] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics (TRO)*, 24(6):1365–1378, 2008.

[83] A. Kawewong, N. Tongprasit, S. Tangruamsub, and O. Hasegawa. Online and incremental appearance-based slam in highly dynamic environments. *The International Journal of Robotics Research (IJRR)*, 2010.

[84] Aram Kawewong, Noppharit Tongprasit, and Osamu Hasegawa. Pirf-nav 2.0: Fast and online incremental appearance-based loop-closure detection in an indoor environment. *Robotics and Autonomous Systems*, 59(10):727–739, 2011.

[85] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP)*, volume 7, 2006.

[86] Sheraz Khan, Athanasios Dometios, Chris Verginis, Costas Tzafestas, Dirk Wollherr, and Martin Buss. Rmap: a rectangular cuboid approximation framework for 3d environment mapping. *Autonomous Robots (AURO)*, pages 1–17, 2014.

[87] Sheraz Khan and Dirk Wollherr. Ibuild: Incremental bag of binary words for appearance based loop closure detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5441–5447, 2015.

[88] Sheraz Khan, Dirk Wollherr, and Martin Buss. Pirf 3d: Online spatial and appearance based loop closure. In *International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 335–340, 2012.

[89] Sheraz Khan, Dirk Wollherr, and Martin Buss. Adaptive rectangular cuboids for 3d mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[90] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A flexible and scalable slam system with full 3d motion estimation. In *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, November 2011.

[91] Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen, Uwe Klingauf, and Oskar von Stryk. Hector open source modules for autonomous mapping and navigation with rescue robots. In *RoboCup 2013: Robot World Cup XVII*, pages 624–631. Springer, 2014.

[92] Kurt Konolige, James Bowman, JD Chen, Patrick Mihelich, Michael Calonder, Vincent Lepetit, and Pascal Fua. View-based maps. *The International Journal of Robotics Research (IJRR)*, 29(8):941–957, 2010.

[93] Kurt Konolige and Willow Garage. Sparse sparse bundle adjustment. In *BMVC*, pages 1–11. Citeseer, 2010.

[94] Miroslav Kulich, Libor Přeučil, et al. Robust data fusion with occupancy grid. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(1):106–115, 2005.

[95] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

[96] Rainer Kummerle, Michael Ruhnke, Bastian Steder, Cyrill Stachniss, and Wolfram Burgard. A navigation system for robots operating in crowded urban environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3225–3232, 2013.

[97] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[98] Sylvain Lefebvre, Samuel Hornus, Fabrice Neyret, et al. Octree textures on the gpu. *GPU gems*, 2:595–613, 2005.

[99] John J Leonard and Hans Jacob S Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *International Symposium on Robotics Research (ISRR)*, volume 9, pages 169–178. Citeseer, 2000.

[100] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011.

[101] Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4372–4378, 2010.

[102] Qingde Li and JG Griffiths. Iterative closest geometric objects registration. *Computers & mathematics with applications*, 40(10):1171–1188, 2000.

[103] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)-Computer Graphics*, volume 21, pages 163–169, 1987.

[104] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-d rigid transformations. In *Proceedings of the 1995 British conference on Machine vision (Vol. 1)*, pages 237–246. BMVA Press, 1995.

[105] Manolis IA Lourakis and Antonis A Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009.

[106] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[107] Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997.

[108] Will Maddern, Alexander D Stewart, Colin McManus, Ben Upcroft, Winston Churchill, and Paul Newman. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Workshop on Visual Place Recognition in Changing Environments, International Conference on Robotics and Automation (ICRA)*, 2014.

[109] M. Magnusson, A. Lilienthal, and T. Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics (JFR)*, 24(10):803–827, 2007.

[110] Martin Magnusson. *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection.* PhD thesis, Örebro University, December 2009. Örebro Studies in Technology.

[111] Martin Magnusson, Andreas Nüchter, Christopher Lörken, Achim J. Lilienthal, and Joachim Hertzberg. Evaluation of 3d registration reliability and speed, a comparison of icp and ndt. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[112] Ruben Martinez-Cantin and Josée A Castellanos. Unscented slam for large-scale outdoor environments. In *IEEE International Conference on Robots and Sytems (IROS)*, pages 3427–3432, 2005.

[113] Zoltan Csaba Marton, Radu Bogdan Rusu, and Michael Beetz. On fast surface reconstruction methods for large and noisy point clouds. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3218–3223, 2009.

[114] Larry Matthies and Alberto Elfes. Integration of sonar and stereo range data using a grid-based representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 727–733, 1988.

[115] Colin McManus, Winston Churchill, Will Maddern, Alexander D Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[116] Colin McManus, Paul Furgale, and Timothy D Barfoot. Towards appearance-based methods for lidar sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1930–1935, 2011.

[117] Robert Mencl and Heinrich Müller. Interpolation and approximation of surfaces from three-dimensional scattered data points. In *IEEE Scientific Visualization Conference (VIS)*, pages 223–223, 1997.

[118] Michael J Milford and Gordon Fraser Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1643–1649, 2012.

[119] Javier Minguez, Luis Montesano, and Florent Lamiraux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics (TRO)*, 22(5):1047–1054, 2006.

[120] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1985–1991, 2003.

[121] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 18, pages 1151–1156, 2003.

[122] Michael Montemerlo and Sebastian Thrun. Fastslam: A scalable method for the simultaneous localization and mapping problem in robotic. *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 63–90, 2007.

[123] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 593–598, 2002.

[124] H Moravec. Robot spatial perceptionby stereoscopic vision and 3d evidence grids. *Perception*, 1996.

[125] Hans Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 116–121, 1985.

[126] Marius Muja and David G Lowe. Fast matching of binary features. In *2012 IEEE Ninth Conference on Computer and Robot Vision (CRV)*, pages 404–410, 2012.

[127] Ana Cris Murillo and J Kosecka. Experiments in place recognition using gist panoramas. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2196–2203, 2009.

[128] Robin R Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan M Erkmen. Search and rescue robotics. In *Springer Handbook of Robotics*, pages 1151–1173. Springer, 2008.

[129] Don Murray and Cullen Jennings. Stereo vision based mapping and navigation for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1694–1699, 1997.

[130] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, et al. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics (JFR)*, 30(1):44–63, 2013.

[131] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, and Yasushi Hada. Redesign of rescue mobile robot quince. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 13–18, 2011.

[132] Takahiko Nakamura and Satoshi Suzuk. Simplified slam using reflection intensity of laser range sensor with retro-reflective marker. In *Annual Conference of the Society of Instrument and Control Engineers (SICE)*, pages 2074–2079, 2012.

[133] Alexandros Nanopoulos, Apostolos N Papadopoulos, and Yannis Theodoridis. R-trees: Theory and applications. *Springer*, 2006.

[134] Tayyab Naseer, Michael Ruhnke, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust visual slam across seasons. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[135] Tayyab Naseer, Luciano Spinello, Wolfram Burgard, and Cyrill Stachniss. Robust visual robot localization across seasons using network flows. 2014.

[136] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International symposium on Mixed and augmented reality (ISMAR)*, pages 127–136, 2011.

[137] P. Newman and K. Ho. Slam-loop closing with visually salient features. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 635–642, 2005.

[138] Paul Newman, David Cole, and Kin Ho. Outdoor slam using visual appearance and laser ranging. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1180–1187, 2006.

[139] Hieu V Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Asian Conference on Computer Vision (ACCV)*, pages 709–720. Springer, 2011.

[140] Viet Nguyen, Ahad Harati, and Roland Siegwart. A lightweight slam algorithm using orthogonal planes for indoor mobile robotics. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 658–663, 2007.

[141] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168, 2006.

[142] A. Nuchter, K. Lingemann, and J. Hertzberg. Cached kd tree search for icp algorithms. In *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 419–426, 2007.

[143] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. Heuristic-based laser scan matching for outdoor 6d slam. In *KI 2005: Advances in Artificial Intelligence*, pages 304–319. Springer, 2005.

[144] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36, 2006.

[145] Edwin B Olson. Real-time correlative scan matching. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4387–4393, 2009.

[146] GK Pandey, James R McBride, Silvio Savarese, and Ryan M Eustice. Toward mutual information based place recognition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3185–3192, 2014.

[147] Kaustubh Pathak, Andreas Birk, Jann Poppinga, and Sören Schwertfeger. 3d forward sensor modeling and application to occupancy grid based sensor fusion. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2059–2064, 2007.

[148] Kaustubh Pathak, Andreas Birk, Narūnas Vaškevičius, and Jann Poppinga. Fast registration based on noisy planes with unknown correspondences for 3-d mapping. *IEEE Transactions on Robotics (TRO)*, 26(3):424–441, 2010.

[149] Rohan Paul and Paul Newman. Fab-map 3d: Topological mapping with spatial and visual appearance. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2649–2656, 2010.

[150] Pierre Payeur, Patrick Hébert, Denis Laurendeau, and Clément M Gosselin. Probabilistic octree modeling of a 3d dynamic environment. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1289–1296, 1997.

[151] Lina M Paz, Juan D Tardós, and José Neira. Divide and conquer: Ekf slam in. *IEEE Transactions on Robotics (TRO)*, 24(5):1107–1120, 2008.

[152] Norbert Pfeifer, Peter Dorninger, Alexander Haring, and Hongchao Fan. *Investigating terrestrial laser scanning intensity data: Quality and functional relations.* 2007.

[153] Norbert Pfeifer, Bernhard Höfle, Christian Briese, Martin Rutzinger, and Alexander Haring. Analysis of the backscattered energy in terrestrial laser scanning data. *International Archives of the Photogrammetry, Remote sensing and Spatial Information Sciences (ISPRS)*, 37:1045–1052, 2008.

[154] Samuel T Pfister, Stergios I Roumeliotis, and Joel W Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 1304–1311, 2003.

[155] Raymond Phan. Equi2cubic. https://github.com/rayryeng/equi2cubic.git, 2011.

[156] Yuval Roth-Tabak and Ramesh Jain. Building an environment model using depth information. *Computer*, 22(6):85–90, 1989.

[157] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 145–152, 2001.

[158] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

[159] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3217, 2009.

[160] Julian Ryde and Jason J Corso. Fast voxel maps with counting bloom filters. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 4413–4418, 2012.

[161] Julian Ryde and Huosheng Hu. 3d mapping with multi-resolution occupied voxel lists. *Autonomous Robots (AURO)*, 28(2):169–185, 2010.

[162] Jari Saarinen, Henrik Andreasson, Todor Stoyanov, Juha Ala-Luhtala, and Achim J Lilienthal. Normal distributions transform occupancy maps: Application to large-scale online 3d mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[163] Jari P Saarinen, Henrik Andreasson, Todor Stoyanov, and Achim J Lilienthal. 3d normal distributions transform occupancy maps: an efficient representation for mapping in dynamic environments. *The International Journal of Robotics Research (IJRR)*, 32(14):1627–1644, 2013.

[164] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[165] Stephen Se, David Lowe, and Jim Little. Local and global localization for mobile robots using visual landmarks. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 414–420, 2001.

[166] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: Science and Systems (RSS)*, volume 2, 2009.

[167] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[168] Gautam Singh and J Kosecka. Visual loop closing using gist descriptors in manhattan world. In *IEEE International Conference on Robotics and Automation (ICRA), Omnidirectional Robot Vision workshop*, 2010.

[169] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477, 2003.

[170] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12):1349–1380, 2000.

[171] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer, 1990.

[172] John P Snyder. *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.

[173] Steven W Squyres, Raymond E Arvidson, Eric T Baumgartner, James F Bell, Philip R Christensen, Stephen Gorevan, Kenneth E Herkenhoff, Göstar Klingelhöfer, Morten Bo Madsen, Richard V Morris, et al. Athena mars rover science investigation. *Journal of Geophysical Research (JGR): Planets (1991–2012)*, 108(E12), 2003.

[174] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, volume 2, pages 65–72, 2005.

[175] Bastian Steder, Giorgio Grisetti, and Wolfram Burgard. Robust place recognition for 3d range data based on point features. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1400–1405, 2010.

[176] Bastian Steder, Michael Ruhnke, Slawomir Grzonka, and Wolfram Burgard. Place recognition in 3d scans using a combination of bag of words and point feature based relative pose estimation. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1249–1255, 2011.

[177] Bastian Steder, Radu Bogdan Rusu, Kurt Konolige, and Wolfram Burgard. Narf: 3d range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE International Conference on Intelligent Robots and Systems (IROS)*, volume 44, 2010.

[178] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[179] Niko Sunderhauf and Peter Protzel. Brief-gist-closing the loop by simple means. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1234–1241, 2011.

[180] Juan D Tardós, José Neira, Paul M Newman, and John J Leonard. Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research (IJRR)*, 21(4):311–330, 2002.

[181] S. Thrun. Learning occupancy grid maps with forward sensor models. *Autonomous robots (AURO)*, 15(2):111–127, 2003.

[182] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[183] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[184] Sebastian Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1:1–35, 2002.

[185] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research (IJRR)*, 23(7-8):693–716, 2004.

[186] Sebastian Thrun and Michael Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research (IJRR)*, 25(5-6):403–429, 2006.

[187] Christian Thurau and Václav Hlavác. Pose primitive based human action recognition in videos or still images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[188] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European Confernce on Computer Vision (ECCV)*, pages 356–369. Springer, 2010.

[189] N. Tongprasit, A. Kawewong, and O. Hasegawa. Pirf-nav 2: Speeded-up online and incremental appearance-based slam in an indoor environment. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 145–152, 2011.

[190] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(9):1226–1238, 2002.

[191] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2276–2282, 2006.

[192] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment, a modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 1999.

[193] Rafael Valencia, Jari Saarinen, Henrik Andreasson, Joan Vallvé, Juan Andrade-Cetto, and Achim J Lilienthal. Localization in highly dynamic environments using dual-timescale ndt-mcl. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3956–3962, 2014.

[194] Matthew Walter, Ryan Eustice, and John Leonard. A provably consistent method for imposing sparsity in feature-based slam information filters. In *Robotics Research*, pages 214–234. Springer, 2007.

[195] J. Weingarten and R. Siegwart. Ekf-based 3d slam for structured environment reconstruction. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 3834–3839, 2005.

[196] Jan W Weingarten, Gabriel Gruener, and Roland Siegwart. Probabilistic plane fitting in 3d and an application to robotic mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 927–932, 2004.

[197] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research (IJRR)*, 34(4-5):598–626, 2015.

[198] Xiao Wu, Wan-Lei Zhao, and Chong-Wah Ngo. Near-duplicate keyframe retrieval with visual keywords and semantic context. In *ACM International Conference on Image and Video Retrieval (CIVR)*, pages 162–169, 2007.

[199] Kai M Wurm, Henrik Kretzschmar, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. Identifying vegetation from laser data in structured outdoor environments. *Robotics and Autonomous Systems*, 62(5):675–684, 2014.

[200] K.M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *ICRA workshop on best practice in 3D perception and modeling for mobile manipulation*, 2010.

[201] Jianxiong Xiao, Krista Ehinger, Aude Oliva, Antonio Torralba, et al. Recognizing scene viewpoint using panoramic place representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2695–2702, 2012.

[202] A. Yahja, A. Stentz, S. Singh, and B.L. Brumitt. Framed-quadtree path planning for mobile robots operating in sparse environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 650–655, 1998.

[203] Brian Yamauchi. Frontier-based exploration using multiple robots. In *ACM International Conference on Autonomous agents (AA)*, pages 47–53, 1998.

[204] Hara Yoshitaka, Kawata Hirohiko, Ohya Akihisa, and Y Shin'ichi. Mobile robot localization and mapping by scan matching using laser reflection intensity of the sokuiki sensor. In *IEEE Annual Conference on Industrial Electronics (IECON)*, pages 3018–3023, 2006.

[205] Alexander Zelinsky. A mobile robot exploration algorithm. *IEEE Transactions on Robotics and Automation*, 8(6):707–717, 1992.

[206] Hong Zhang. Borf: Loop-closure detection with scale invariant visual features. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3125–3130, 2011.

[207] Li Zhang and Bijoy K Ghosh. Line segment based map building and localization using 2d laser rangefinder. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2538–2543, 2000.

[208] Robert Zlot and Michael Bosse. Place recognition using keypoint similarities in 2d lidar maps. In *International Symposium on Experimental Robotics (ISER)*, pages 363–372. Springer, 2009.

Own Publications

[209] Martin Buss, Daniel Carton, **Sheraz Khan**, Barbara Kühnlenz, Kolja Kühnlenz, Christian Landsiedel, Roderick de Nijs, Annemarie Turnwald, and Dirk Wollherr. Iuro–soziale mensch-roboter-interaktion in den straßen von münchen. *at-Automatisierungstechnik*, 63(4):231–242, 2015.

[210] Nikos Mitsou, **Sheraz Khan**, Eirini Ntoutsi, Dirk Wollherr, Costas Tzafestas, and Hans Peter Kiegel. An uncertain 3d grid formulation for indoor environment mapping. In *International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016.

[211] **Sheraz Khan**, Athanasios Dometios, Chris Verginis, Costas Tzafestas, Dirk Wollherr, and Martin Buss. Rmap: a rectangular cuboid approximation framework for 3d environment mapping. *Autonomous Robots*, 37(3):261–277, 2014.

[212] **Sheraz Khan**, Nikos Mitsou, Dirk Wollherr, and Costas Tzafestas. An optimization approach for 3d environment mapping using normal vector uncertainty. In *International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 841–846, 2012.

[213] **Sheraz Khan** and Dirk Wollherr. Ibuild: Incremental bag of binary words for appearance based loop closure detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5441–5447, 2015.

[214] **Sheraz Khan**, Dirk Wollherr, and Martin Buss. Pirf 3d: Online spatial and appearance based loop closure. In *IEEE International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 335–340, 2012.

[215] **Sheraz Khan**, Dirk Wollherr, and Martin Buss. Adaptive rectangular cuboids for 3d mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[216] **Sheraz Khan**, Dirk Wollherr, and Martin Buss. Modeling laser intensities for simultaneous localization and mapping. *IEEE Robotics and Automation Letters (RA-L)*, 2016.

[217] Dirk Wollherr, **Sheraz Khan**, Christian Landsiedel, and Martin Buss. The interactive urban robot iuro: Towards robot action in human environments. In *Experimental Robotics*, pages 277–291. Springer, 2016.