Simultaneous localization and map building for outdoor mobile robot navigation

Zhang, Sen

2006

Zhang, S. (2006). Simultaneous localization and map building for outdoor mobile robot navigation. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4124>

<https://doi.org/10.32657/10356/4124>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:54:36 SGT

4438538

Simultaneous Localization and Map Building for Outdoor Mobile Robot Navigation

Zhang Sen

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirements for the degree of

Doctor of Philosophy

2006

TJ
211.415
Z63
2006

Statement of Originality

I hereby certify that the content of this thesis is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

31, May, 2006

Date

Zhang Sen

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Xie Lihua, for his professional guidance, generous supports, encouragement and invaluable advices throughout the past three years. His insightful comment and thoughtful discussion has been an inspiration for my research. It has been a great pleasure to have Professor Xie as my advisor.

I also give my sincere thanks to Associate Professor Martin David Adams, for his co-supervision of my studies, fruitful discussions and suggestions. Many thanks also go to Associate Professor Wang Danwei for his kind help.

I would also like to thank all my colleagues, friends and the technicians in the Computer Control Laboratory. Their support, encouragement and friendship will always be treasured. I would like to acknowledge the School of Electrical and Electronic Engineering, Nanyang Technological University for providing the financial support, research facilities and opportunity for this study.

Finally, my deep thanks go to all the members of my family. My deepest gratitude, in love and affection, belongs to my parents, for their understanding, encouragement and strong supports. Particularly I thank my parents looking after my daughter during this difficult period. And I have to say sorry to my lovely daughter that I spent too little time with her during these three years. I save the last special thanks for my husband who stands by me through the most difficult time of this process, offering his unconditional love and support.

Summary

Simultaneous Localization and Map Building (SLAM) is currently an active research area in autonomous ground vehicles (AGVs). The SLAM problem asks if it is possible for an autonomous vehicle to start at an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute absolute vehicle location. Solving this problem would allow an AGV to be deployed easily, with very little initialization requirement and enable it to be truly autonomous that can deal with changes in the environment.

In order to navigate in an unknown environment, an AGV is often faced with some fundamental difficulties: natural landmark (feature) extraction, matching of extracted landmarks and computational requirement in real time implementation of SLAM algorithms.

This thesis is firstly concerned with the feature extraction problem for indoor mobile robot navigation. For semi-structured outdoor environments, we propose two algorithms to extract useful features. One is an optimization based algorithm. The algorithm consists of data segmentation and parameter acquisition. It applies the extended Kalman filter (EKF) for data segmentation and a modified Gauss-Newton method for fitting circle parameters iteratively. The other is a switch model approach. By establishing appropriate planar and circle models, EKF and unscented Kalman filter (UKF) based tracking methods are developed where switching between models are carried out based on statistical tests. Experi-

mental results demonstrate that the proposed feature extraction methods can give more accurate feature parameters than existing methods.

For more complex outdoor environments than the semi-structured (walls and trees are the main features) environments, we present another feature extraction algorithm. In this approach, a model based on a change in curve gradient is developed to segment data from various types of environments and determine feature types. Experimental results in a real outdoor environment show that the proposed algorithm performs very well.

Data association is another critical problem of the SLAM process. Correct matching of the observed features to map landmarks is very important for consistent map building because a single false match manyinvalidate the entire process [1]. In this thesis, an efficient data association approach is presented for high feature density (the distance between features is small) environments. In this approach, the feature based SLAM data association problem is formulated as a 0-1 integer programming (IP) problem using one frame scan data. An iterative heuristic greedy rounding procedure is applied to solve the IP problem. Detailed simulation and real time experimental studies show that the proposed approach offers better performance than the Nearest Neighbor (NN) method especially when the feature density is high. In the simulations, it is also compared with the work in [2, 3] and the joint compatibility branch and bound (JCBB) method.

The implementation of any SLAM algorithm in large outdoor areas is usually difficult because of the computational and memory resources needed. In order to reduce the computational burden in SLAM, we propose an entropy based feature selection strategy. The algorithm, though not optimal, can select features that could provide good estimation of vehicle pose and features' positions based on the availability of computing resources. The simulation and experimental results show that the proposed entropy based feature selection strategy is effective in reducing computational burden for the SLAM in a large scale environment.

Table of Contents

Acknowledgments	i
Summary	ii
List of Figures	xiii
List of Tables	xv
1 Introduction	1
1.1 Background and Motivation	1
1.1.1 Early Localization Systems	2
1.1.2 Beacon Based Localization	3
1.1.3 A Prior Map-based Localization and Map Building	4
1.1.4 Simultaneous Localization and Map Building (SLAM)	7
1.2 Objective	10
1.3 Major Contributions of the Thesis	10
1.4 Organization of The Thesis	12
2 Issues in SLAM and Literature Review	15
2.1 Introduction	15

TABLE OF CONTENTS		v
2.2	Metric Representation Based Approaches for SLAM	16
2.2.1	Occupancy Grid Based SLAM	16
2.2.2	Feature Based SLAM	18
2.3	Feature Extraction	19
2.4	Data Association	23
2.5	Computational Burden in SLAM	25
2.6	Conclusions	27
3	Feature Based Simultaneous Localization and Mapping System	29
3.1	Introduction	29
3.2	EKF Based SLAM Algorithm	30
3.3	Map Management	34
3.4	Sensors	36
3.4.1	Odometry (Encoders)	37
3.4.2	Laser Measurement System (SICK LMS)	38
3.4.3	Summary of Advantages of Laser Sensor	40
3.5	Statistics of Laser Sensor Noise	41
3.6	Cycab AGV	42
3.6.1	Vehicle Model	44
3.6.2	Sensor Model	46
3.7	Conclusions	47
4	Feature Extraction Algorithms for SLAM in Outdoor Environ- ments	48
4.1	Introduction	48

TABLE OF CONTENTS

vi

4.2 Gauss-Newton Feature Extraction Algorithm

50

4.2.1 Data Segmentation and Edge Detection

51

4.2.2 Parameter Acquisition

54

4.3 Feature Extraction via Model Switching

57

4.3.1 Cylindrical Surface (Circle) Model

57

4.3.2 Model Switching

62

4.4 Curve Gradient Model Based Feature Extraction Approach

64

4.5 Simulation and Experimental Results

68

4.5.1 Results of Gauss-Newton Feature Extraction and Model Switch-
ing Feature Extraction

68

4.5.2 Results of Curve Gradient Model Based Feature Extraction .

82

4.5.3 Discussions on the Proposed Algorithms

92

4.6 Conclusions

93

5 An Efficient Data Association Approach for SLAM

97

5.1 Introduction

97

5.2 Problem Formulation

99

5.3 The IHGR Based Data Association Algorithm

102

5.3.1 Gating

102

5.3.2 Iterative Heuristic Greedy Rounding

103

5.3.3 Algorithm Complexity

105

5.4 Simulation and Experimental Results

106

5.4.1 Results of Simulation Environment

106

5.4.2 Real Outdoor Environment

116

TABLE OF CONTENTS

vii

5.4.3	Discussions on the Proposed Algorithm	121
5.5	Conclusions	125
6	Entropy-based Method of Feature Selection	126
6.1	Introduction	126
6.2	Entropy-based Feature Selection for SLAM	128
6.2.1	Stochastic Mapping	128
6.2.2	Problem Formulation	129
6.3	Entropy Based Feature Selection Strategy	132
6.4	Map Management	135
6.5	Simulation and Experimental Results	136
6.5.1	Simulation Results	136
6.5.2	Experimental Results	141
6.6	Conclusions	144
7	Conclusions and Recommendations for Further Research	146
7.1	Conclusions	146
7.2	Recommendations for Further Research	148
	Author's Publications	149
	Bibliography	151
	Appendices	168

List of Figures

2.1 Example of occupancy map in [4] (white - free space, black - obstacle; blue - unexplored space) 17

3.1 SLAM estimation process 30

3.2 Map management 35

3.3 LMS 290-S01 39

3.4 The laser scan 39

3.5 The operating principle 40

3.6 The range noise distribution for black objects in the daytime (the noise mean $\mu_r = -0.0095m$ and the standard deviation $\sigma_r = 0.0167m$). 42

3.7 The range noise distribution of laser sensor for black objects at night (the noise mean $\mu_r = -0.0193m$ and the standard deviation $\sigma_r = 0.0178m$). 42

3.8 The range noise distribution for white objects in the daytime (the noise mean $\mu_r = -0.0151m$ and the standard deviation $\sigma_r = 0.0184m$). 43

3.9 The range noise distribution of laser sensor for white objects at night (the noise mean $\mu_r = -0.0247m$ and the standard deviation $\sigma_r = 0.0142m$). 43

3.10 The Cycab, a car-like vehicle in our experiment 44

3.11 The Cycab’s Kinematic Model 45

3.12 The observation relationship between the vehicle and the features
in the global coordinate 47

4.1 The relationship between successive range readings when scanning
a planar surface 52

4.2 The relationship between successive range readings when scanning
a cylindrical surface 58

4.3 The relationship between successive range readings when scanning
a cylindrical surface ($\alpha = 0$) 59

4.4 A gradient model of a smooth surface for edge detection. 65

4.5 The place to be explored by the robot. 70

4.6 Data from One whole scan. 71

4.7 Circles and edges extracted from data scanned at position 2 using
the proposed Gauss-Newton method (the normal view). 71

4.8 A zoomed view of the region inside box A in Figure 4.7. 72

4.9 A zoomed view of circles and edges extracted from data scanned at
position 4 using Gauss-Newton method. 72

4.10 Circles and edges extracted from data scanned at position 2 using
the model switching method (the normal view). 73

4.11 A zoomed view for the region inside box B in Figure 4.10. 73

4.12 The circles and edges extracted from data scanned at position 4 by
Gauss-Newton method (the normal view). 74

4.13 A zoomed view of the region inside box C in Figure 4.12. 74

4.14 Circles and edges extracted from data scanned at position 4 using
the model switching approach (the normal view). 75

4.15 A zoomed view of the region inside box D in Figure 4.14. 75

4.16 The second experimental environment (the whole scene). 79

4.17 The estimated path and the true trajectory of the vehicle during the SLAM. 80

4.18 A whole scan data of the second experiment environment corresponding to Figure 4.16. 80

4.19 The circle features (trees) and edges extracted from the environment in Figure 4.16 using Gauss-Newton algorithm. 81

4.20 A comparison on range and bearing innovations during the localization when using the proposed method (dash-dot line) and the method in [5] (solid line in the middle). 81

4.21 The error and 3σ error bounds of the vehicle when using different feature detection methods. The dashed line is the result of the localization using the proposed feature detection methods. 82

4.22 The scan data of the simulation environment and the feature extraction results 83

4.23 Map of the experimental environment. 86

4.24 Scan1038 87

4.25 Scan1056 87

4.26 Scan1066 88

4.27 Scan1076 88

4.28 The features and the comparison of the truth path and the estimated trajectory in the dynamic environment. 89

4.29 The range observation innovation during the localization process. . 90

4.30 The bearing observation innovation during the localization process. 90

4.31 The map of our experiment site. 92

4.32 A typical laser scan during the vehicle moving. 93

4.33 The features (circle features and edges) extracted from the zoomed view of the block E in Figure 4.32. 94

4.34 The map and the estimated path and GPS reading of the path during the SLAM. In the figure, the third feature (from the start point of the trajectory) is at (17.8,-26.7) and its covariance is $\sigma_{xx} = 0.1748; \sigma_{yy} = 0.0955; \sigma_{xy} = \sigma_{yx} = 0.0146$ 95

5.1 The unsuccessful mapping when applying the NN data association algorithm in the SLAM process. 107

5.2 The mapping and vehicle path when applying IHGR data association method in the SLAM process. 109

5.3 The vehicle’s position errors in the global coordinate between the ground truth (generated data) and the estimated position using IHGR data association algorithm in the SLAM process. 109

5.4 The mean execution time of the IHGR algorithm and the NN algorithm. The mean execution time means the average CPU time used of 50 Mote Carlo runs for the process of the data association algorithm. The mean execution time of IHGR is nearly linear with respect to the observed feature number by repeated experiments. . . 110

5.5 The SLAM path and the feature map during the SLAM process with IHGR data association. 117

5.6 The range innovation and its 2σ confidence bounds during the SLAM process with IHGR data association. 117

5.7 The observation angle innovation and its 2σ confidence bounds during the SLAM process with IHGR data association. 118

5.8 The map of our experiment site. The blue curve represents the vehicle’s trajectory; The starting and ending of the trajectory are indicated by red dots. 121

5.9 The environment and the vehicle in our experiment. 122

5.10 A typical laser scan of the experimental environment. 122

5.11 A comparison between the estimated path and the path by the GPS readings during SLAM. 123

5.12 The range observation’s innovation and its 95% confidence bounds in the SLAM experiment. 123

5.13 The angular observation’s innovation and its 95% confidence bounds in the SLAM experiment. 124

5.14 The unsuccessful path estimation of this experiment when NN algorithm is applied. The vehicle started from point A in our experiment, see Figure 5.8. 124

6.1 The mapping and vehicle path when using the proposed entropy based feature selection in the SLAM process. 137

6.2 The vehicle’s position errors (between the ground truth (GPS reading) and the estimated position) comparison in the global coordinate between the entropy based feature selection in the SLAM process and full SLAM. 137

6.3 The range innovation and its 2σ confidence bounds during the SLAM process with the proposed method. 138

6.4 The angle innovation and its 2σ confidence bounds during the SLAM process with the proposed method. 138

6.5 Part of the experimental site (we started from this point). 142

6.6 The real map of the vehicle trajectory at Nanyang Technological University. 142

6.7 The experimental results. The figure shows the vehicle’s estimation path (red line), the features extracted (green circles) and the selected features (red star) by using the entropy based feature selection algorithm during the SLAM process. 143

6.8 The experimental results. The range innovation and its 3σ confidence bounds during the SLAM process with the entropy based SLAM. 143

6.9 The experimental results. The observation angle innovation and its 3σ confidence bounds during the SLAM process with the entropy based SLAM. 144

List of Tables

4.1 The summed square errors $E(p)$ of the four circle features using the proposed methods 76

4.2 A comparison of the error CE of the four circle features between the two proposed methods and the method in [5] 77

4.3 A comparison of the error RE of the four circle features between the two proposed methods and the method in [5] 77

4.4 The $\Delta x, \Delta y, \Delta \theta$ of the vehicle pose when using different feature detection methods 78

4.5 A comparison of the relative error of the circle features and edge features between the proposed method and the method in [6] 84

5.1 The comparison of the SLAM performance for various number of observations (M). 110

5.2 SLAM performance versus the standard deviation of the sensor range noise. 112

5.3 The comparison of the SLAM performance for the JCBB algorithm and IHGR algorithm under different number of features. The unit for the errors is meter. 112

5.4 The typical scan matching results. 113

5.5 The comparison of the features' tracking loss rate of several data association methods. 114

5.6 The comparison of the computational time of several data association methods. 114

5.7 The comparison of the features' tracking loss rate of several data association methods under different feature densities. 115

5.8 The comparison of the computational time of several data association methods. 115

5.9 The matching matrix in scan 68. 119

5.10 The matching matrix in scan 87. 119

6.1 The information difference for each observation in Scan 437. 139

6.2 SLAM performance and the algorithm computational complexity versus the number of the selected features. 140

Chapter 1

Introduction

1.1 Background and Motivation

Accurate localization is arguably the most fundamental competence required for autonomous ground vehicle (AGV) systems. It forms the basis for most navigation and control decisions. Without accurate localization, an AGV is essentially left to wander its environment with no notion of where it is and where it is going. All actions that an AGV is likely to perform depend on its accurate localization. For example, path planning in an environment requires continuous knowledge of the vehicle location. In certain applications of AGVs, such as underwater exploration [7], the failure of localization may even lead to the loss of the AGV. The importance of AGV navigation has been recognized for long time, and in the last decade much research effort has been expended to achieve autonomous navigation. Several kinds of methods have been presented by which accurate localization of a vehicle can be achieved. In this thesis, we focus on one of the AGV systems—mobile robots in field environments.

1.1.1 Early Localization Systems

Line following robots in manufacturing industry may be seen as the birth of AGV navigation systems. Line following is the process of guiding an AGV along a fixed path using a track that the AGV can identify and follow. This is a simple navigational method that requires only very simple sensing and processing resources [8]. Line following methods have been implemented over long distances. An example is the Rapidly Adapting Lateral Position Handler (RALPH) of the Carnegie Mellon University [9], which uses curbs and lane markings to keep a vehicle within the lane of a road. This system has successfully guided a vehicle from New York to Los Angeles. However, approaches that rely on line following have significant drawbacks. A major disadvantage is that the mobile robot can only follow a line, and therefore its operation is severely restricted. A further problem of line following strategies is the cost and time needed to lay tracks for the AGV. Line following strategies are also inflexible since considerable new investment in capital and time is needed to modify the path of the vehicle.

An alternative navigation method that does not suffer from the problems outlined above relies on the localization of the AGV. Localization differs from line following in that it attempts to specify the pose of the vehicle within the environment rather than make small corrections on its motion. A limited form of localization is possible using odometry or an inertial navigation system (INS) [10, 11]. Odometry (generally the use of wheel and steer encoders) is an important part of mobile robot navigation, particularly in the case of wheeled and tracked vehicles. Essentially, counting wheel rotations provides good short distance accuracy and is inexpensive and amenable to a high sampling rate. Inertial measurement units (IMUs), that use accelerometers and gyros to measure acceleration and rate of rotation, are increasingly being used for mobile robot navigation. In an IMU, measurements are integrated (twice for accelerometer and once for rotation rate) to obtain the position and attitude of a vehicle. However, both odometry and INS suffer from the

accumulation of errors during their operations, and therefore lead to unbounded localization errors [11].

1.1.2 Beacon Based Localization

Beacon based navigation systems are commonly used in navigation and navigation-aiding for both indoor and outdoor vehicles. In beacon based localization systems, a set of beacons is placed at known positions of an environment and the AGV is equipped with an external sensor (such as laser range sensor) that can observe the beacons. The observation information of the beacon positions is applied to localize the vehicle. Unlike odometry and INS, the accuracy of the localization does not deteriorate with time in a beacon based system. The reason for this is that the external sensor (laser range sensor in our case) on the AGV provides observations of the position of the AGV relative to the stationary beacons which are used to update the AGV's pose, rather than only measuring observations of the self-motion of AGV by inertial sensors, and therefore is not subjected to the accumulation of integration errors. Beacons may be passive or active. Passive beacons include radar reflectors used in the navigation of cargo or mining vehicles, or laser-reflectors used in indoor navigation systems [12, 13]. Active beacons may include radio transponders in outdoor applications, or acoustic methods in indoor and sub-sea environments.

Probably the best-known beacon-based localization system is the Global Positioning System (GPS) [14, 15]. GPS based localization relies on both the signals emitted by several GPS satellites and the time the signals travel. Essentially, the GPS receiver compares the time when a signal is transmitted by a satellite with the time when it is received. Knowing the position of the satellites and the time-of-flight of the signal obtained, the position of the GPS receiver on the AGV is determined (The AGV's pose is determined). Beacon-based systems are often used in conjunction with odometric or inertial dead-reckoning sensors which also

provide navigation information when GPS is blocked to form an overall navigation system. The principal advantage of using GPS over land based beacons is that the GPS signal is already available and therefore it is not necessary to install beacons. This reduces the deployment cost and deployment time of the system.

A common problem of beacon based localization systems is that beacons have to be installed and the beacons may get damaged or obscured, compromising the performance of the AGV. This may be expensive and may also increase the deployment time of the AGV.

1.1.3 A Prior Map-based Localization and Map Building

Localization

Map based navigation is an extension of beacon-based navigation methods to the case where natural landmarks or terrain maps are used in place of artificial beacons [16]. The landmarks (also considered as naturally occurring beacons or features) are extracted from successive sensor observations. Map-based localization compares the observed information by the sensors and the information of the map. If a Kalman filter technique is used to calculate the vehicle pose, the differences between the predicted features' locations and the location observations of the features are then applied to provide real-time updates of vehicle position with bounded position error.

There has been a significant amount of work on map-based localization and a number of methods have been proposed for achieving accurate localization given maps of the environment within which the vehicle operates [17, 18, 19, 20, 21, 22]. It is difficult to say which one is best. It depends on the necessary accuracy, the cost, the environment where the vehicle operates, etc. In general, good results can be obtained when a set of sensors (with different physical principles) are integrated.

The integration of the information coming from the set of sensors is usually done by means of sensor fusion. The fusion takes into account the noise probability distributions and the mathematical models that relate the measured data and the states to be estimated.

Several authors have used triangulation approaches for mobile robot localization. Triangulation is a well-known technique for finding a position by means of bearings from fixed points in the environment [23, 24, 25]. For example, Esteves [26] addresses a version of the localization problem where a geometric triangulation algorithm is used for the self-localization of a robot on a plane. The robot measures the direction of rays each of which emanates from the robot's position.

For other methods, recent work by Thrun [27, 28] has shown that accurate, robust, real-time localization can be achieved in indoor environments using grid based and Monte-Carlo methods to estimate the position of a vehicle given a known map of an indoor environment. These applications have typically shown good results in small scale, indoor environments. Similar methods have also been applied to large scale outdoor environments [29]. The literature abounds with other examples of mobile robotic systems that use a variety of sensors, including lasers, sonars and cameras, to perform accurate localization.

Localization approaches can also be distinguished by the kind of data available to the robot. Dead reckoning has always been attractive because of its pervasiveness [30, 31]. With the rapid development of technology, GPS receivers are quickly becoming the sensor of choice for outdoor localization [32, 33]. Laser range finders have also become very popular mobile robot components [34, 35].

Map Building

An autonomous vehicle must have the ability to build and maintain maps of initially unknown environments. Learning map with mobile robots has received con-

siderable attention over the last two decades. This is because maps are often inherently necessary for mobile robots to perform their tasks, and they can also be used for future navigation purposes. In cases where the robot has good information about its localization, it is not difficult to create a map based on the information obtained from its external sensors.

Maps allow the vehicle to plan its movement within the environment in order to achieve its goals. The automatic creation of maps allows the vehicle to model the world using its sensors installed on it. The map building problem is concerned with the identification of previously observed and unobserved features, types of features and integration of features into a map. It is usually assumed that the location of the vehicle is precisely known during the map building phase. Landmark observations that cannot be classified as previously observed features generate new features in the map. Consequently, the data association problem also lies at the heart of the map building problem [36]. The maps are used as a tool for performing other higher-level tasks.

A key issue in the practical implementation of mapping schemes concerns how map information is represented, processed, stored, updated, and retrieved. A number of map building techniques rely on the extraction of relevant environmental features [36, 37].

From a mapping point of view, the first important problem to be considered is the type of information that represents the environments. Many systems use a geometrical representation of the navigation space that is defined as feature based maps where the 2D or 3D physical space is mapped according to the absolute geometric relationships between objects. The level of detail in feature maps varies. Some detailed account of the feature-based method will be covered in the next Chapter.

The occupancy grid technique divides the covered region into lots of cells. Each cell has a value that represents the probability of this cell being occupied. There is a

compromise between the size of the cells (one of the factors to improve the quality of the map) and the complexity to manage a big number of cells. The feature based maps and the occupancy grid based maps constitute the first category of maps—the metric map approach.

The second category—topological maps represent the environment based on topological relationships between features using a graph or a net. Because the topological approach relies on geometric information, it can be seen as a higher resolution representation of a geometric map, and is mostly used for path planning purposes in small scale environments.

The third category, which combines the advantages of both of the above approaches, is hybrid metric-topological map. In this case, the map is connected by distinctive objects (nodes) so as to form a topological map and at the same time each node can be represented by the metric representation. Because a global topological map connects local metric maps, allowing a compact environment model in the map, it does not require global metric consistency and permits both precision and robustness.

1.1.4 Simultaneous Localization and Map Building (SLAM)

In unknown and generally unstructured environments, it is desirable to combine both map building and localization functions, that is, simultaneous localization and map building (SLAM), also known as CML (concurrent mapping and localization). The SLAM algorithm is considered as a good choice for the robot navigation as it does not require a priori map and thus makes the robot adaptive to the change of the environments. It has the ability to localize a mobile robot in an unknown environment and simultaneously build up a map and use this map to navigate. In the SLAM process, the poses of landmarks are stored in the map and updated by their new observations from sensors mounted on the robot. During the update

process, the robot pose estimate is used to update the landmarks' positions and at the same time the landmarks are used to improve the estimate of robot's pose. Because the landmarks are repeatedly observed, the map will tend to converge after long time update.

The problem of SLAM is a significant open problem in mobile robotics which is difficult because of the following paradox: to localize itself, the robot needs the map of the environment, and, for building a map the robot localization must be known precisely. In other words, the SLAM problem is hard because the same sensor data must be used for both mapping and localization.

This problem has been studied in the past two decades by different researchers [38, 39, 40, 41, 42, 43]. Many researchers have successfully extended SLAM from indoors [44, 45] to outdoors [46], from on land [47, 48] to underground [49, 50] and underwater [51, 52] environments, and from 2D to 3D [53, 54].

The first consistent solution to the SLAM problem was published by Smith, Self and Cheeseman [55]. Although they did not examine the SLAM effects, their solution provides a consistent treatment of SLAM. This method, here referred to as stochastic SLAM, remains the basic foundation of all subsequent practical SLAM proposals using the landmark-based map framework. Early experimental verification of the algorithm with laser [56] and sonar [57] sensors demonstrated its utility in relatively small-scale indoor environments. Modifications of the approach have also been published [58]. Leonard and Durrant-Whyte present an important work in SLAM [59]. Their method uses sensor information to define features and the relation among them to direct a robot experiencing positioning error. Through the proper understanding of the robots relationship to the features, the robot can maintain an accurate estimate of its position while moving through the environment using the features.

For practical SLAM in large-scale unstructured environments, there are still several main open problems that cannot be solved perfectly until now: feature detection,

data association and computational burden, etc. Our motivation comes from these existing issues. We shall give their details as follows:

- Feature extraction: Only when the environment information can be represented suitably and extracted precisely, the vehicle's pose estimation has a higher probability to be correct. An incorrect feature extraction of the observation can cause the filter for localization divergent, effectively rendering all future predicted observations incorrect. How to represent the outdoor environments and what kind of features should be used is one of the main problems in SLAM and also in navigation. There still lack of good and systematic methods to solve it. This has motivated us to investigate methods for this problem.
- Data association: Data association is a very important problem for SLAM. Correct correspondence between data obtained from robot sensors and the data currently stored in the map is essential for SLAM. A data association algorithm should be robust to both a high feature density situation and an absence of stable features. The classical method for data association is the Nearest Neighbor(NN) method which cannot give a satisfactory result in environments that contain many closely distributed features. Recently developed approaches are mostly based on an optimal searching algorithm which is not very efficient for real time implementation.
- Computational burden: It refers to that when the size of the operating environment is increased, the complexity of correlations between vehicles to features and features to features is increased. This leads to heavy computational overhead. When the number of features is large, real time implementation becomes difficult.

1.2 Objective

Our research is focused on the feature-based SLAM in outdoor environments. The feature-based SLAM is to enable a mobile robot to build a map of an unknown environment while concurrently using that feature-based map to navigate. The objective of this work is to develop algorithms for feature extraction, data association and computational requirement problem and build a reliable real-time localization system for mobile robot in outdoor environments which can realize the developed algorithms. As mentioned before, for practical SLAM in large-scale unstructured environments, there are still several main open problems that cannot be solved perfectly until now. Focusing on these problems, we summarize the objective as follows:

- To develop new feature extraction algorithms for outdoor mobile robot SLAM.
- To develop an efficient data association algorithm for outdoor environments of high feature density with a performable computational requirement.
- To investigate methods for reducing the computational burden of SLAM algorithm.
- To integrate the hardware and software developed for our AGV system in order to implement the proposed algorithms in real outdoor environments.

1.3 Major Contributions of the Thesis

Major contributions from the study can be summarized as follows:

1. *Two new feature extraction algorithms for semi-structured outdoor environments for mobile robot navigation are developed.* For semi-structured outdoor environments such as campuses and parks, commonly encountered natural

features that can be very useful for mobile robot navigation include edges (large discontinuity) and circles (eg, trees, pillars). Two new methods are developed in the study. The first algorithm is a batch processing approach which carries out feature extraction after measurements from a full scan are received. The algorithm consists of data segmentation and parameter acquisition. A modified Gauss-Newton method is proposed for fitting circle parameters iteratively. In the second algorithm, a switch model based approach is developed for tracking features of different geometries such as edges, circles and random clutters by applying EKF and unscented Kalman filter (UKF).

2. *A curve gradient model for smooth surfaces is proposed for data segmentation and determination of feature type.* In order to generalize the proposed Gauss-Newton feature extraction method into more complex outdoor environments, a curve gradient model is developed. It is applicable to various types of features including lines, arcs and circles.
3. *A data association approach is developed based on the integer programming (IP) technique.* Correspondences between the sensed feature observations and map landmarks are essential for consistent map construction since any single false matching may invalidate the entire process. A data association algorithm should be robust to both high feature density situation and an absence of stable features. We present an efficient IP based data association approach to SLAM. Unlike the traditional NN algorithm, the proposed algorithm deals with the global matching between existing features and measurements of each scan and is more robust for an environment of high feature density (the feature number is high and the distances between features are often very close) which is usually the case in outdoor applications. Detailed simulation and experimental studies show that the proposed iterative heuristic greedy rounding (IHGR) based algorithm offers a better performance for complex environments of high feature density than the popular NN algorithm. Compared to other algorithms, such as JCBB and the work in [2, 3], it has lower computational burden.

4. *In order to reduce the computational burden, we propose an entropy-based method for feature selection in SLAM for outdoor mobile robot navigation.* We implement a feature selection strategy which is based on information theory (entropy) together with a map management method to initialize new features into the map, match measurements to the map features and delete the out-of-date features. The selected features are sub-optimal in the sense that fusion of measurements from those features with existing information would yield good entropy reduction of the robot location and the map features' locations. Our method has the advantage of selecting a suitable number of features by considering the computational constraint in real time implementation. The simulation results show that the proposed entropy based feature selection strategy is effective in reducing the computational burden in SLAM.
5. *Real time implementation.* We implement software and hardware combination for our autonomous vehicle. All the algorithms proposed above are implemented for real time outdoor experiments.

1.4 Organization of The Thesis

The thesis is organized as follows:

Chapter 2 firstly gives a comprehensive literature review of SLAM. The crucial issues of SLAM are also summarized.

Chapter 3 introduces the basics of the feature based SLAM systems. We firstly give the EKF based SLAM algorithm, then describe the sensors and vehicle used in this study.

Chapter 4 studies the problem of feature extraction for outdoor environments such as campuses and parks using laser range sensors. Here we first present two new algorithms for feature extraction. One approach applies the EKF for data

segmentation and a modified Gaussian-Newton optimization algorithm for parameter acquisition. The other approach is based on model switching. Simulation and experimental results show that the proposed two algorithms are implementable in detecting features for semi-structured outdoor environments where lines and circle features are the main features.

This chapter also develops a curve gradient model for smooth surfaces in order to segment sensor data for more complex outdoor environments. It can be applied not only for line segmentation in semi-structured environments, but also for curves in less structured environments. The most notable feature of the curve gradient model is its ability in adapting to data recorded from arbitrary surfaces, such as building walls, trees, or even complex irregular surfaces. The effectiveness of the feature extraction method is verified by simulations and experiments.

Chapter 5 presents an efficient integer programming (IP) based data association approach to SLAM. In this approach, the feature based SLAM data association problem is formulated as a two dimensional assignment problem. Different from the work in [2, 3], we consider one frame scan data, which avoids the NP hard optimization problem in [2, 3] where multiple frame scans are dealt with. In our algorithm, all possible hypothesis in the validation gate for the measurements and existing features are considered. The assignment problem is approached by first solving a relaxed linear programming (LP) problem. Based on the optimal LP solution, a suboptimal solution to the IP problem is then obtained by applying an IHGR procedure. Both the simulation and experiment results demonstrate that the proposed algorithm is implementable and gives a better performance than the NN algorithm. Comparisons with the optimal JCBB algorithm [60] and the algorithm in [2, 3] are also given.

Chapter 6 studies the problem of how to reduce computational burden for SLAM. In order to reduce the computational requirement of SLAM for real time implementation, only a limited number of features which can provide higher information

gain for the estimation of the vehicle's pose and features' position are selected. An entropy based suboptimal feature selection method is proposed that aims to maximize the total knowledge about the system in the presence of measurement and navigation uncertainty. The number of the features to be selected depends on both the available computational resource and the accuracy requirement of the estimation. Simulation and experimental results are given to show the performance of the feature selection scheme for outdoor SLAM.

Chapter 7 concludes the thesis and provides some possible future research directions.

Chapter 2

Issues in SLAM and Literature Review

2.1 Introduction

Autonomous vehicles are expected to adapt to the ever changing environment with minimal user intervention. The assumption of a prior map by many localization algorithms is not acceptable for a mobile robot. The SLAM builds a map concurrently when the mobile robot estimates its own position in the initially empty map. Its implementation should deliver a good answer to the challenging localization requirement of autonomous robots. This thesis focuses on SLAM and its related techniques for mobile robots in outdoor environments. In order to have a complete understanding of the SLAM problem, in this chapter, we firstly review the existing metric based approaches of stochastic mapping, namely the grid occupancy and the feature based stochastic mapping, then the main research issues of SLAM and their existing techniques are introduced.

2.2 Metric Representation Based Approaches for SLAM

Most of the algorithms suggested for localization, mapping, and SLAM problems have one common feature: they are probabilistic. They employ stochastic equations for modelling robot systems and uncertainties arising in the interactions between the robot and its environment. For SLAM, the representation of the environment is often discussed by using topological representations [61] and metric ones [62, 63]. Techniques based on topological maps have less requirements in terms of modelling and computational needing, whereas metric maps can provide a more precise localization. Our study focuses on the latter. Thus we will only give the details of the metric representation for SLAM in the following.

There are two commonly used metric representation approaches for SLAM—occupancy grid based SLAM and feature (or landmark) based SLAM. In this section, we will review the history and the state-of-the-art of these two methods.

2.2.1 Occupancy Grid Based SLAM

Historically, the maps were represented as occupancy grids in some early work—a 2D array of cells (see Figure 2.1 for an example of an occupancy map [4]). Each cell is assigned a value that represents the probability that it is occupied by an obstacle. In Figure 2.1, black represents that the corresponding cells are occupied with obstacle in probability one while white represents clear space. The grey color region means that the probability the region is occupied by an obstacle is between 0 and 1.

After the pioneering work of Giralt et al. [64] and Moravec and Elfes [4], this approach was adopted by many researchers [65, 66, 67]. Most of the systems based on occupancy maps use sonar or laser sensors [68, 69], some other works are on

vision sensors such as [70]. Although these works are not directly on SLAM, they formed a foundation for the recent work of SLAM that is based on the occupancy grid technique.

Figure 2.1: Example of occupancy map in [4] (white - free space, black - obstacle; blue - unexplored space)

As we know, the occupancy grid mapping approach has some difficulties when it is used for large outdoor mobile robot localization. The most significant difficulty with occupancy grids is the tradeoff between grid resolution (granularity) and computational burden in large environments. For SLAM, besides this difficulty, another main problem with occupancy grid maps is that they do not possess an appropriate uncertainty model for sensor and vehicle pose uncertainty and do not take into account the correlation between cells that exists due to the robot pose uncertainty. Therefore this method will tend to diverge in large outdoor environments if it is not combined with other techniques when it is used for SLAM.

Recently, there were several works on SLAM that are based on the occupancy grid technique combined with other approaches [71, 72, 73]. In [72], the SLAM algorithm (named DP-SLAM) applies the occupancy map to represent the environment rather than the landmarks. The DP-SLAM (version 2.0) addresses the problem of how to eliminate the computational burden when the particle filter technique

and the occupancy grid map are applied to SLAM at the same time. The DP-SLAM algorithm makes the occupancy grid based SLAM practical in real outdoor environments and offers a promising solution of occupancy grid based SLAM for environments where it is difficult to extract useful features, such as terrain. In [73], the authors presented the Hybrid Metric Maps (HYMMs) framework for the representation of environments for SLAM. The HYMMs combines feature maps with occupancy grid information to represent the environment where the robot explores. This approach takes the advantages of both the feature based map and the occupancy grid map for SLAM to produce a detailed and consistent environment representation for the implementation of DenseSLAM defined by the authors.

2.2.2 Feature Based SLAM

Conventional SLAM techniques are generally based on feature map representation of the environments. Feature maps (landmark maps) in SLAM [74] represent the environment by the global locations of parametric features, such as lines, circles, corners or points.

The feature based SLAM involves fusing observations of features or landmarks with dead-reckoning information to track the location of the robot in the environment and build up a map of landmark locations at the same time. In the SLAM process, the sensor measurements' uncertainty contributes to the uncertainty of the vehicle pose estimation and the uncertainty of the features position estimation. These uncertainties are correlated in the form of a covariance matrix when the EKF is applied to update the system state of the SLAM process.

For the EKF based SLAM, in order to guarantee consistency of the feature map, all the correlations (mentioned above) among landmarks and vehicles have to be maintained. This requires that the estimates of the vehicle pose and the feature locations be stored in the same state vector which will be augmented when there

are new features observed and added to the map. With the augmentation of the state vector, the update process becomes slower and slower. Therefore, for large outdoor environments, improvements to the basic feature extraction, data association and stochastic SLAM algorithms are required in order to obtain a feasible real time implementation of feature based SLAM. For small-scale environments, if stable features are observed, feature based SLAM performs well enough in most situations.

2.3 Feature Extraction

Feature-based localization (including SLAM) is performed by extracting features from sensed data and associating them with features in the map firstly. The core of the localization problem is the reliable acquisition or extraction of information about navigation landmarks from sensor information and the automatic correlation or correspondence of these with some navigation map [75].

Therefore, feature extraction is one of the most important issues for feature-based SLAM. The performance of the mobile robot localization depends firstly on the ability of obtaining information from its sensors.

In outdoor navigation, the features used include two types: natural landmarks and artificial beacons. For most of the outdoor navigation systems, the natural landmarks are more popular than the artificial beacons. There are several advantages in using natural landmarks for navigation purpose over artificial navigation beacons. For example, the radar beacons used are constructed from simple metal surfaces, even these must be manufactured, installed and maintained. Several navigation systems rely on active beacons that are able to relay additional identification information back to the vehicle [76]. Hence, the system complexity and cost is higher. The second reason is that the environments are often dynamic and frequent changes in structure are common. Mine sites, where minerals are constantly

extracted and new areas are developed are a good example of such a dynamic environment. With a fixed beacon infrastructure, additional objects must be surveyed to cover the new areas of operation. The vehicles must also be updated, with the changes to the beacon grid, so they may localize successfully in all areas. All of this is time consuming and thus costly.

Therefore, in many outdoor environments, natural features are used for SLAM. Popular natural feature for SLAM includes point features, poles, trees, walls, and so on. Similarly to human and other animal behavior, a robot can consider these types of remarkable objects before it tries to use a more general and complete description of the environment [77].

Hence, fast and reliable algorithms capable of extracting features from a large set of noisy data are very important as the absence of simple features leads to the need for more complex perception and modelling. Some of the early efforts in this direction have focused on extracting line features in an indoor environment based on the information provided by sonar and laser sensors. In [78], a least-squares line fitting technique was applied to extract edges from ultrasonic sensor data. In [79], a recursive line fitting system is used to extract line segments under polar coordinates and an ellipse fitting method is also implemented for data from a laser sensor. In [80], line segments are detected using a regression least-squares parameter estimation method whereas the center and radius of a circle feature are estimated based on the average value of the measurements of the circle from a 2D range scanner. Instead of fitting straight line segments after a full scan has been recorded, Adams presented an on-line edge extraction approach employing a Kalman filter in [81]. Later, based on this method, a two-layer Kalman filter is used to calculate the parameters of a line by an on-line means in [82]. Observe that the aforementioned works are focused on indoor applications and are mainly concerned with line extraction.

For an outdoor environment, the problem of feature selection and detection is

more challenging. For outdoor environments, any natural feature that is chosen as a mark for SLAM has to satisfy the following important criteria, see also [83]:

- A feature should be re-observable in the explored environment. There are often the cases that the features observed in some early observations are blocked by other objects between the robot and the features. In this case, the features cannot be re-observed. Thus the confidence in features' location and vehicle's location decreases and the accuracy of the estimation of the vehicle and other features will tend to be low. This may cause the failure of the SLAM algorithm.
- The number of features should be enough in certain region. If there are few features that can be re-observed for SLAM, the uncertainty of the estimated location will rapidly increase. Thus it is necessary to have enough features to maintain the update process of SLAM. However, the actual number of features necessary is difficult to be decided for different situations. It depends on some factors including the vehicle model accuracy, vehicle velocity, the sensor characteristics etc.
- The features should be predominant. This means there are almost no feature clusters formed around that feature. If the number of feature clusters around a feature is high, the probability of wrong matching between the observation and the feature will also be high and this may result in failure of the SLAM process.
- The feature types used should be as few as possible. The popular natural features for SLAM include point features, lines, corners, circles, and so on. The multiple types of the features require multiple hypothesis [84, 85] and multiple filters. This requires some heuristic approaches to lessen the computational complexity. Therefore a less number of feature types is preferable for the real time implementation of SLAM.

In our view, in most typical semi-structured outdoor environments, such as cam-

pushes, parks and suburbs, tree trunks and tree-like objects, such as pillars, are relatively stable, regular and naturally occurring features that can provide very useful information for mobile robot navigation. Recently, some research on the use of these kinds of geometrical features has been carried out in [86, 5]. Also, references [86, 5] address the problem of extracting tree trunks from laser scan data where the centre and radius of a circle are estimated by averaging the measurements. This method can be susceptible to outliers which can significantly affect the accuracy of the center and radius estimates. Adams presented an on-line gradient based surface discontinuity detection method by using outdoor scanning range sensors [6]. In this approach, a range gradient model is derived for outdoor complex surfaces in order to detect large discontinuities in these surfaces. Only edges can be detected in this approach. Other relevant works include graphics for circle detection, see, for example, Philippe Dosch [87] where an arc detection algorithm for vision data is presented. The approach is based on recursive splitting to segment a curve into a set of arcs and segments via least-squares minimization. Also Doris proposed a vector-based arc segmentation in [88]. However, in [87, 88], there lack of systematic methods in determining some parameters that are needed in their algorithms and the computational requirement of these algorithms is high. Different from the above methods, Madhavan et al. [89] presented a scale space method to extract the points of maximum curvature from laser range scans of unmodified outdoor environments. This method is effective for point feature detection rather than other type features. Recently, another adaptive smoothing algorithm using a modified scale space method by Tang et al. [90] also gives a reliable solution for point feature extraction by employing a line model mask to smooth range data and preserve the edges at different scales.

2.4 Data Association

SLAM also relies on correct correspondence between data obtained from the robot sensors and data currently stored in the map. Data association failure is a serious problem for SLAM. For simple localization, mis-matching of extracted features may not lead to failure of the localization process because the features's pose is known. However, for SLAM, wrong matching will affect the features' map and this can lead to divergence of the SLAM process. Data association is a crucial step during SLAM.

There have been numerous approaches to data association.

In stochastic mapping, the simplest method is the NN algorithm which is a classical technique in tracking problems [91]. The great advantage of NN is its $O(mn)$ computational complexity in addition to its conceptual simplicity (Here m is the number of sensor measurements and n is the number of existing features in the map). It performs satisfactorily when clutter density is low and sensor accuracy is high. However, during the process of SLAM, especially in complex outdoor environments, clutter level is usually high and the innovations in matching different observations obtained from the same vehicle position are correlated. In this situation, the NN algorithm may accept a wrong matching, which leads to divergence in state estimation.

In order to improve the robustness of data association, Neira and Tardós [60] present an approach using a joint compatibility test based on the branch and bound search with a high computational cost. Juan Nieto *et al.* [92] give a fast SLAM algorithm for data association by applying the multiple hypotheses tracking method in a variety of outdoor environments. The experimental complexity estimates show that if the number of features in one scan is large, these algorithms will not be fast enough for real time implementation. In other approaches, Bailey *et al.* [93] consider relative distances and angles between points and lines in two laser

scans and use graph theory to find the largest number of compatible pairings between the measurements and existing features. The work of Lim and Leonard [94] applies a hypotheses test to implement data association of the relocation in SLAM using geometrical constraints. Castellanos and Tardós [95] use binary constraints to localize the robot with an *a priori* map using an interpretation tree. In these methods, geometric constraints among features are used to obtain hypotheses with pairwise compatible pairings. However, pairwise compatibility doesn't guarantee joint compatibility [60], and additional validations are required.

Other data association algorithms used in multi-target tracking include joint probability data association (JPDA) and multiple hypotheses tracking (MHT) [96]. The JPDA associates all the measurements that are falling inside the validation region of a track to itself by a probabilistic weighting procedure and performs fairly well in highly cluttered environments, but the calculation of the weighting probability is difficult in a SLAM application. Historically, the MHT algorithm firstly proposed by Reid [97] has been considered to be the only approach that can truly provide an optimal data association solution. However, its practicality and feasibility have been hampered since it requires the enumeration of an exponentially increasing number of feasible joint association hypothesis to evaluate probabilities. Similar to MHT, another data association method that is formulated as a discrete optimization problem (0-1 integer programming) was firstly proposed in [98]; Later, it was improved by A.B. Poore et al. in [99]. In this method, in order to solve the multidimensional assignment problem, known to be an NP-hard problem, more efficiently, Lagrangian relaxation approach is applied. This data association method can use more than one previous frame of measurements to determine the best solution for the current matching. In [100], a linear programming based approach was presented to reduce the computational burden for the multidimensional assignment problem of data association. Recently, by extending the idea of multi-sensor multi-target tracking problem, Perera et al. [2, 3] proposed a promising approach for SLAM data association. In this approach, the SLAM data

association problem is formulated as a three dimensional assignment problem by using two frame sensor measurements. It establishes the difference among false alarms, new feature measurements and missed detections in order to obtain an effective data association scheme.

2.5 Computational Burden in SLAM

SLAM is a much more challenging task compared to the other localization problems. In very large scale environments, the computation and memory requirements of SLAM can be very expensive. This is caused by the high number of the estimated variables of the SLAM system. High computation and storage are the price paid for maintaining correlations in the state covariance matrix. In the SLAM problem, the state of landmarks of the map and the state of the vehicle pose form the system state to be estimated. The more the number of the landmarks, the higher the state dimension. Denoting the dimension of the state vector by n , an $n \times n$ covariance matrix needs to be updated when one new observation is used to update the system state. The computational complexity is $O(n^2)$ [101]. In general, the number of the feature parameters is more significant compared to the number of the vehicle pose parameters.

If the SLAM period is long, the number of landmarks will increase and after some time the computer resources will not be sufficient to update the map in real time. Real-time performance becomes impossible for environments with more than a few hundred features.

There have been some approaches to address the computational problem. In [102] it is assumed that the error in the AGV estimate is not correlated with the errors in the landmark estimates. Recently some new methods have been developed. Disanayake et al. [103] proposed a method of removing redundant map features to reduce computational requirement. The corresponding map management problem

of this SLAM method is addressed in [104]. Leonard et al. [101, 105] presented a method which splits the map into multiple globally-referenced submaps, each with its own vehicle track, and maintains all correlations within a submap. In [63], the authors proposed a Constrained Local Submap Filter (CLSF) which relies on creating an independent, local submap of the features in the immediate vicinity of the vehicle. And the local submap is then periodically fused into the global map of the environment. In [106, 107], authors also divided the environment into a network of submaps. The motivation is to achieve good performance by computing multiple partial solutions in parallel, and to avoid the computational burden that is entailed by computing one complete solution. They also presented an approximation technique to address the update of the covariance in the transition between the submaps. A heuristic algorithm is used to propagate the vehicle estimates between submaps. Although impressive experimental results are given, there is no proof of consistency of the approach or estimation of the conservatism of the covariance over-bounding strategy. Newman has analyzed the convergence properties of the covariance matrices in SLAM and developed techniques for the scaling problem based on the use of relative maps [108, 109]. It was shown that the computational costs vary between being linear in the number of features and the nearly optimal solution. However, there are two difficulties with this method. One is that the estimates of beacons are not expressed in the same consistent coordinate frame and must be converted using projection operators [108]. The other problem is that the map cannot be used to directly update the vehicle. Simon Julier and Uhlmann introduced a covariance intersection (CI) for state estimate of SLAM [110]. The computational cost is linear with the number of beacons [110, 111]. But the method for SLAM implemented with CI may not provide an effective solution because the error bounds can be too conservative. Guivant and Nebot presented a compressed filter that allows the accumulation of measurements in a local area with a cost proportional to the square of the number of the landmarks in this area independent from overall map size n [112, 113, 114]. When the robot leaves this region, the accumulated results must be propagated to the global update. In this

approach, the number of times that the full $\sim O(n^2)$ update must be performed in the local map has to be reduced.

Montemerlo et al. [115] used a particle filter to track the pose of the robot, where each particle also includes a set of Kalman filters estimating the position for each landmark. This approach is able to represent and search between multiple hypotheses for the full map (i.e., robot pose plus all landmark positions), but the particle set must be large enough to contain a particle sufficiently close to the true pose of the robot at all times. The algorithm requires $O(m)$ complexity for m particles, though it is not clear how the number of particles scales with the complexity of the environment. Thrun et al. [116] presented a constant time algorithm called Sparse Extended Information Filter (SEIF). They followed a similar idea proposed by Frese and Hirzinger [117] and used information matrix instead of covariance matrix to represent uncertainty. The algorithm in [116] exploits that the information matrix is approximately sparse, requiring $O(kn)$ storage space (where k is the number of landmarks seen by the robot in each local region). The number k of these landmarks influences the computation time of the algorithm. It depends on the sensor and the density of landmarks but does not grow when the map gets larger. However, the states estimated by IF have to be transformed from the information space to the system (vehicle and features) state spaces. Due to the need of calculating the Jacobian matrices in the linearization process for the localization purpose, the transformation has at least a cost proportional to n^2 .

2.6 Conclusions

This chapter introduced some background information and fundamental issues of SLAM. It gave some detailed review of the metric map representation. Their relative merits and utility for SLAM were also discussed. The chapter also reviewed three crucial problems of SLAM: feature extraction, data association and compu-

tational requirement of SLAM.

Chapter 3

Feature Based Simultaneous Localization and Mapping System

3.1 Introduction

As stated before, SLAM is a very complex and difficult problem in the mobile robot localization field. In order to have a complete understanding of the nature of the SLAM process, in this chapter, we shall describe the stochastic mapping process of SLAM based on EKF, a map management method, and also the characteristics of sensors and vehicle that are used throughout the thesis.

The chapter firstly presents the mathematical formulation of the SLAM problem. The estimation of vehicle pose and feature positions (mapping) are carried out using the well-known EKF algorithm. In Section 3.2, mathematical models for the vehicle, the environment and the sensors are described. Section 3.3 examines the map management problem in SLAM. Section 3.4 describes our SLAM system, sensors and vehicle used. Section 3.5 investigates the noise characteristics of laser sensors. Section 3.6 introduces the vehicle's model and the conclusions are drawn in Section 3.7.

3.2 EKF Based SLAM Algorithm

The EKF is a standard estimation tool used widely in the construction of navigational systems. A brief discussion of estimation using EKF is firstly introduced [118, 112].

Figure 3.1: SLAM estimation process

The Vehicle and Feature Models

The SLAM problem is often performed by starting a vehicle from an unknown location. The vehicle moves through an environment which contains a set of features whose locations are also unknown. The vehicle is equipped with encoders and external sensors. The external sensors are used to measure the relative distance between any individual feature and the vehicle itself. The vehicle motion in any real navigation problem is often expressed as follows:

$$\mathbf{x}_v(k+1) = \mathbf{f}_v(\mathbf{x}_v(k), \mathbf{u}(k)) + \mathbf{v}_v(k) \quad (3.1)$$

where

$$\mathbf{x}_v(k) = \begin{bmatrix} x_v(k) \\ y_v(k) \\ \theta_v(k) \end{bmatrix}$$

with x_v and y_v being the coordinates of the vehicle and θ_v the heading angle. $\mathbf{f}_v$ defines the kinematic model of the vehicle with $\mathbf{u}(k)$ being the vector of control inputs. $\mathbf{v}_v(k)$ is a white noise with zero mean and covariance $\mathbf{Q}_v$.

The position of the i th feature is denoted by p_i . The system state equation for the i th feature is

$$p_i(k+1) = p_i(k)$$

The features are assumed to be stationary and the number of features in the environment is assumed to be N . Thus the augmented state equation of the vehicle and all features is expressed as follows:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k) \quad (3.2)$$

$$\text{where } \mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_v(k) \\ p_1(k) \\ \vdots \\ p_N(k) \end{bmatrix}, \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_v(k), \mathbf{u}(k)) \\ p_1(k) \\ \vdots \\ p_N(k) \end{bmatrix}, \mathbf{v}(k) = \begin{bmatrix} \mathbf{v}_v(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The Observation Model between the Vehicle and the Features

The external sensor that is installed on the vehicle can get observations of the relative positions between features and the vehicle. The observation model of the i th feature is expressed as follows:

$$\mathbf{z}_i(k) = \mathbf{H}_i(\mathbf{x}_v(k), p_i(k)) + \mathbf{w}_i(k) \quad (3.3)$$

where $\mathbf{w}_i(k)$ is a white noise with zero mean and variance σ_r . The observation function $\mathbf{H}_i(\cdot, \cdot)$ gives the relationship between the sensor measurement and the

system state variable when observing the i th feature.

The SLAM Estimation Process with EKF

We apply the EKF for the state estimation of the vehicle and landmarks. The flow chart of the estimation process can be seen in Figure 3.1. Given the estimate $\hat{\mathbf{x}}(k | k)$ of $\mathbf{x}(k)$ and control input $\mathbf{u}(k)$, the predicted state $\hat{\mathbf{x}}(k + 1 | k)$ using (3.2) is given by

$$\hat{\mathbf{x}}(k + 1 | k) = \mathbf{f}(\hat{\mathbf{x}}(k | k), \mathbf{u}(k)). \quad (3.4)$$

The prediction error covariance is approximately given by:

$$\mathbf{P}(k + 1 | k) = \mathbf{F}(k)\mathbf{P}(k | k)\mathbf{F}^T(k) + \mathbf{Q}(k) \quad (3.5)$$

where $\mathbf{F}(k)$ is the transition matrix of equation (3.2) after the linearization. $\mathbf{P}(k | k)$ is the prior error covariance estimation at time k . $\mathbf{Q}(k)$ is the covariance of the white noise $\mathbf{v}(k)$, i.e. $\mathbf{Q}(k) = \text{diag}\{\mathbf{Q}_v(k), 0, \dots, 0\}$.

In view of (3.3), the predicted measurement is simply

$$\hat{\mathbf{z}}_i(k + 1) = \mathbf{H}_i(\hat{\mathbf{x}}_v(k + 1 | k), \hat{p}_i(k + 1 | k)) \quad (3.6)$$

where $\hat{\mathbf{x}}_v(k + 1 | k)$ and $\hat{p}_i(k + 1 | k)$ are the elements of $\hat{\mathbf{x}}(k + 1 | k)$ which is calculated from equation (3.4). Then, the difference between the measurement and the predicted observation, namely the innovation, is given by

$$\nu_i(k + 1) = \mathbf{z}_i(k + 1) - \hat{\mathbf{z}}_i(k + 1). \quad (3.7)$$

Thus, the covariance of the innovation is:

$$\mathbf{s}_i(k + 1) = \nabla \mathbf{H}_i(k + 1)\mathbf{P}(k + 1 | k)\nabla \mathbf{H}_i(k + 1)^T + \sigma_r^2 \quad (3.8)$$

where $\nabla \mathbf{H}_i(k+1)$ is the Jacobian matrix of the observation function with respect to the predicted system state $\hat{\mathbf{x}}(k+1 | k)$. Because each observation is only a function of the feature being observed, the matrix is a sparse matrix of the form:

$$\nabla \mathbf{H}_i(k+1) = [\nabla_v \mathbf{H}_i(k+1) \quad \mathbf{0} \quad \dots \quad \mathbf{0} \quad \nabla_i \mathbf{H}_i(k+1) \quad \mathbf{0} \quad \dots] \quad (3.9)$$

where $\nabla_v \mathbf{H}_i(k+1)$ and $\nabla_i \mathbf{H}_i(k+1)$ are the Jacobians of the observation function with respect to the vehicle states and the i th feature state, respectively. The EKF gain is given by

$$\mathbf{K}_i(k+1) = \mathbf{P}(k+1 | k) \nabla \mathbf{H}_i(k+1)^T \mathbf{S}_i^{-1}(k+1). \quad (3.10)$$

At time $k+1$, we use new matched observations one by one in the current sensor information (current scan in our application) to update the estimate using the following equations:

$$\hat{\mathbf{x}}_1^- = \hat{\mathbf{x}}(k+1 | k), \quad (3.11)$$

$$\hat{\mathbf{P}}_1^- = \hat{\mathbf{P}}(k+1 | k), \quad (3.12)$$

$$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \mathbf{K}_i(k+1) \nu_i(k+1), \quad (3.13)$$

$$\hat{\mathbf{P}}_i^+ = \hat{\mathbf{P}}_i^- - \mathbf{K}_i(k+1) \mathbf{s}_i(k+1) \mathbf{K}_i^T(k+1), \quad (3.14)$$

$$\hat{\mathbf{x}}_{i+1}^- = \hat{\mathbf{x}}_i^+ \quad (i = 1, \dots, n), \quad (3.15)$$

$$\hat{\mathbf{P}}_{i+1}^- = \hat{\mathbf{P}}_i^+ \quad (i = 1, \dots, n), \quad (3.16)$$

$$\hat{\mathbf{x}}(k+1 | k+1) = \hat{\mathbf{x}}_n^+, \quad (3.17)$$

$$\mathbf{P}(k+1 | k+1) = \hat{\mathbf{P}}_n^+ \quad (3.18)$$

where i means the i th observation in the current scan and n is the number of observations in the current scan; $\hat{\mathbf{x}}_i^-$ is the system state estimate before update using the i th observation and $\hat{\mathbf{x}}_i^+$ is the estimate after the update by observation i . $\hat{\mathbf{P}}_i^-$ and $\hat{\mathbf{P}}_i^+$ are the corresponding state covariance matrices, respectively.

3.3 Map Management

If a map is to be built incrementally as information is gathered from sensors, there is typically a need for a map management process. This process has the function of managing the information present in the knowledge base, adapting the map representation to specific application requirements, and possibly aiding the sensing process. Given the fact that computational resources are limited, an information management technique that reduces the stored data without sacrificing much information is required. To improve the applicability of a spatial description to a larger variety of scenarios, it should present the ability to iteratively adapt its geometry to application-specific requirements.

The map management process can be divided into three aspects for SLAM in dynamic environments as follows [119, 104]:

1. Adding feature. When a feature observed in the current scan cannot be matched to the existing feature list, a new feature is initialized.
2. Removing redundant features. If all static features are included for updating the state, the computational requirement will be high. Thus, redundant features that have not been observed for a long time interval should be removed. Measures for assessing feature reliability and information content will be discussed in Chapter 6.

Figure 3.2: Map management

3. Removing unstable features. Features become unstable or obsolete if they move or become permanently occluded. For example, features might be stationary for a long period of time, and can be considered suitable features for SLAM. But if they move, they are unstable features and should be removed from the map. Another case is that structural changes may occur in the environment—such as trees or poles removed. Other cases, such as, an object might be placed in front of a feature, occluding it from view. For whatever reason, some features may cease to exist and no longer provide useful information. These unstable features should be deleted from the map.

In Figure 3.2, a flow chart for the map management of SLAM is shown. We can see that, after data association, if a feature cannot be matched to any existing feature in the map, it is considered as a new feature. The feature initialization is activated. Otherwise, this observation is used for the map update.

After a specified time interval, we shall check if this feature is still matched by any new coming observations during this period. If it is matched by none of the observations sensed from external sensors within the specified interval, this feature should be removed from the feature listing. Otherwise, this feature will still be kept in our map.

Finally, the sensing process can be improved if sensors are told where to look at. This directed sensing technique will naturally have benefits, such as to speed up the map building process, or to extract information about the environment in a predefined way.

3.4 Sensors

For almost every land navigation application we can always find an appropriate combination of dead reckoning sensors and other sensors that can be used to ob-

tain a reasonable prediction of the trajectory of a vehicle. Different sensors have different shortcomings.

For outdoor applications, GPS is widely used but its accuracy cannot be guaranteed when the satellite signal is blocked by, for example, trees or buildings. Ultrasonic sensors have been applied in indoor environments, but they are not adequate for outdoor situations due to the range limitation and bearing uncertainties. Stereo vision seems to be a promising technology, but its complexity and poor dynamic range make it unreliable for outdoor applications. Millimeter wave radar has enormous potential but it is very expensive.

Recently, laser sensors have become one of the most attractive sensors for localization purpose due to their accuracy and low cost [75]. Most laser sensors provide bearing and range information with sub degree of resolution and an accuracy of the order of 1-10 cm and a range of up to 80 meters. In the past several years, the use of laser range measurement systems for autonomous navigation tasks has been on the rise [75, 120, 121, 122, 123]. However, the predominant domain of their use has been in indoor environments [120, 121, 122, 123]. In our study, we focus on outdoor environments. We shall explore and implement the SLAM by applying the laser scanner together with GPS and encoders on the wheels of the vehicle. Therefore, we firstly give an explanation about the sensors used.

3.4.1 Odometry (Encoders)

The word odometry comes from the Greek words for road and measure [11] and is the name of the most used technique for making a robot find out its position. Odometry works by integrating incremental information over time. By using wheel encoders to count the number of revolutions of each wheel, the robot measures the distance it travelled and its heading direction. The new position is calculated using the encoders on the wheel motors and knowing some physical dimensions of the

robot (wheel diameter and wheel distance).

Odometry is widely used, because it gives good short-term accuracy, is inexpensive, and allows for very high sampling rates [11]. However, due to drift and slippage the integration of the wheel revolutions leads to errors in both travelled distance and orientation [11]. These errors accumulate over time. In particular errors in the orientation cause large positioning errors. Another disadvantage of odometry is its sensitivity to terrain. If the surface the robot is driving on is not smooth, it can result in considerable position errors, since the odometry system cannot detect the irregularities in the terrain. Also differences in wheel diameter can cause position errors that are not detected by odometry measurements [10].

Although odometry causes increasing error in the location estimate, it is the most accessible form of position information and therefore it is an important source of information for localization. In our study, the vehicle's position is estimated by odometry together with laser range scanner so as to get an accurate result.

3.4.2 Laser Measurement System (SICK LMS)

The LMS (Laser Measurement System) 290-S01, see Figure 3.3, is a divergent laser scanner with a maximum scanning angle of 180° and a lateral resolution of between 0.25° and 1° (variable and definable). Also, the laser scanner carries out a planar scan. The accuracy of measurement in a single shot is about ± 2 cm. With respect to the data transfer rate, the LMS is capable of transferring all measured values in real-time (via serial interface) at a baud rate of 500 Kbauds.

As seen from Figure 3.4, the LMS does a planar scan of 180° with an angular resolution choice of 0.25° , 0.5° or 1° . This implies that for a 180° scan with 0.5° angular resolution, there will be a total of 361 scan values. Each scan value consists of 2 bytes of data and the LMS will send the number of data bytes according to the request. In our use, the angular resolution of the laser scanner is 0.5° .

Figure 3.3: LMS 290-S01

Figure 3.4: The laser scan

Operating Principle

The operating principle of our non-contact LMS is based on time-of-flight measurement. Please see Figure 3.5. The LMS calculates the distance to the object using the time of flight of pulsed light, i.e. the length of time between sending and receiving the beam of light.

An extremely short pulse of light (infrared laser beam) is transmitted towards an object. Part of the light is reflected back to the unit after a fraction of a second. A rotating mirror deflects the pulsed light beam to many points in a semi-circle. The precise direction is given by an angular sensor on the mirror (laser RADAR). A large number of coordinates measured in this way are put together to form a model of the surrounding area's contours. Using the serial interface of the unit, measurements are transferred in real time (or other specified rates) to a host PC for further evaluation.

Figure 3.5: The operating principle

3.4.3 Summary of Advantages of Laser Sensor

There are many advantages of the laser scanner:

- It is very fast. In most cases, the measurements can be considered to be instantaneous. This means that we do not have to think about the compensation for the motion of the platform between sending and receiving. In a sonar system, most of time is wasted, waiting for the pulse to return. When using laser, the problem is instead to be able to rotate the scanning device fast enough and at the same time to obtain accurate ranging.
- The angular resolution of laser sensor is far better than that of the other sensors, such as sonar and radar.
- The range accuracy is fairly good. The new generations of laser sensor of SICK Electri-Optics usually has an accuracy of better than 10mm and an angle resolution of 0.25° .
- The laser data can be interpreted directly as the range from the obstacle to the sensor. But for a camera, this takes a lot of effort to interpret.

3.5 Statistics of Laser Sensor Noise

The laser sensor used in our project has been shown in Figure 3.3. In the following, we shall analyze the statistics of the laser sensor.

A number of experiments have been conducted in which 2000 samples were taken at each target position and the standard deviations of the depth measurement noises of each target were computed.

In the experiments, we have examined how standard deviation changes when the laser beam light scans on surfaces of different materials of the target. We placed a target at a known distance, took 2000 samples, and computed the standard deviation and the mean of the range noise. From Figure 3.6 to Figure 3.9, we can see that under different illumination conditions (day and night) and using different targets the Sick sensor noise changes slightly. In the figure, the gaps that appear

Figure 3.6: The range noise distribution for black objects in the daytime (the noise mean $\mu_r = -0.0095m$ and the standard deviation $\sigma_r = 0.0167m$).

Figure 3.7: The range noise distribution of laser sensor for black objects at night (the noise mean $\mu_r = -0.0193m$ and the standard deviation $\sigma_r = 0.0178m$).

in the histograms are due to the discrete nature of the time-of-flight measurement.

The histograms in the figures have horizontal axes showing the measured range noise and vertical axes showing the number of readings which have a particular noise reading (using 500 bins in Matlab command hist). All of the histograms in the figures were produced from different targets or the same target under different illumination conditions at a fixed range, 20.00 meters, in this case, from the sensor. We calculate the noise for each sample of the range and get 2000 noise samples at each case. The histogram is labelled with the noise sample mean and standard deviation. Other experiments have also been done for different distance from 5 meters to 20 meters, but the results are similar. The noise statistics change slightly. Note that the distributions in the figures are almost Gaussian.

3.6 Cycab AGV

The Cycab in Figure 3.10 is a small intelligent vehicle that runs on electricity. It had been designed to transport up to two persons in downtown areas, pedestrian malls, large industrial or amusement parks and airports, at a maximum speed of

Figure 3.8: The range noise distribution for white objects in the daytime (the noise mean $\mu_r = -0.0151m$ and the standard deviation $\sigma_r = 0.0184m$).

Figure 3.9: The range noise distribution of laser sensor for white objects at night (the noise mean $\mu_r = -0.0247m$ and the standard deviation $\sigma_r = 0.0142m$).

30 km/h.

It offers two functions, aided driving and autonomous driving. Aided driving relies on the use of a joystick and a finger touch screen. This joystick is connected to the computer, which controls the vehicle and then provides secure and easy driving. In addition, speed may be limited in curves and special areas like narrow lanes, to improve safety of the vehicle. The control of the Cycab is simple and user-friendly and anyone will be able to drive this vehicle without any special skills. The finger touch screen allows the user to communicate with the system in order to get information such as localization, vehicle autonomy, or even a list of nearby restaurants. One possible application of the Cycab is to form a “virtual car train” of empty vehicles with only one driver in the first vehicle (also known as Car Following). Other autonomous driving modes, like radio-control or light markers guidance, may be implemented.

The Cycab has been designed with mass production and public use constraints in mind. The considering factors are low cost, compactness, robustness and easy maintenance. Some parameters and information about Cycab can be found in Appendix A.

Figure 3.10: The Cycab, a car-like vehicle in our experiment

3.6.1 Vehicle Model

An accurate vehicle model is an essential component of most navigation schemes. Depending on the reliability with which the motion of the vehicle can be modeled and the sensors available for predicting the change of state of the vehicle, highly accurate estimates of vehicle motion can be generated.

A vehicle model attempts to capture the fundamental relationship between the vehicle's future state, $\mathbf{x}_v(k+1)$, and its current state, $\mathbf{x}_v(k)$, given a control input, $\mathbf{u}(k)$. Specifically, the state equation is given by: $\mathbf{x}_v(k+1) = \mathbf{f}_v(\mathbf{x}_v(k), \mathbf{u}(k)) + \mathbf{v}_v(k)$, where $\mathbf{x}_v(k)$ is the state vector of the vehicle at time instant k and $\mathbf{v}_v(k)$ is a random vector describing both the dynamic driving noise and uncertainties in the state-space model of the vehicle. In the implementation, we use the Cycab which is a typical car-like vehicle (see Figure 3.10). We make the basic assumptions of planar motion, i.e. the vehicle is a rigid body and there is no slippage of tire. The status of the vehicle in motion can be described in the global coordinates by using the "bicycle model". To describe the vehicle motion, a global coordinate X-Y is

fixed on the horizontal plane on which the vehicle moves. The motion status of the vehicle at an arbitrary moment can be described using the Bicycle Model as illustrated in Figure 3.11.

Reference point G is chosen at the center of gravity of the vehicle body. In order to obtain a common vehicle model (see Equation (3.19)), the mass center G is selected as the origin of the local coordinate system. (x_v, y_v) represents the position of the vehicle; Vehicle velocity v is defined at the reference point G ; Heading angle θ_v is the angle from the X -axis to the longitudinal axis of the vehicle body; Side-slip angle β is the angle from the longitudinal axis of the vehicle body to the direction of vehicle velocity, v ; Turning radius r is the distance between the reference point G and the Instant Rotating Center O ; Front wheel velocity v_f is the velocity defined at the intersection of the mid-plane of the virtual front wheel and the front wheel axle; Rear wheel velocity v_r is the velocity defined at the intersection of the mid-plane of the virtual rear wheel and the rear wheel axle; Front wheel steering angle δ_f is the angle from the longitudinal axis of the vehicle body to the direction of v_f .

Figure 3.11: The Cycab's Kinematic Model

The vehicle model is as follows [124]:

$$\begin{bmatrix} x_v(k+1) \\ y_v(k+1) \\ \theta_v(k+1) \end{bmatrix} = \begin{bmatrix} x_v(k) + \Delta \cdot v(k) \cos(\theta_v(k) + \beta(k)) \\ y_v(k) + \Delta \cdot v(k) \sin(\theta_v(k) + \beta(k)) \\ \theta_v(k) + \Delta \cdot \frac{v(k)}{l_r} \sin \beta(k) \end{bmatrix} + \mathbf{v}_v(k) \quad (3.19)$$

where Δ is the sampling interval, v and β are the control inputs that are computed from encoder increments during the interval Δ . The above can be put into the general form of (3.1) with

$$\begin{aligned} \mathbf{f}_v(\mathbf{x}_v(k), \mathbf{u}(k)) &= \begin{bmatrix} x_v(k) + \Delta \cdot v(k) \cos(\theta_v(k) + \beta(k)) \\ y_v(k) + \Delta \cdot v(k) \sin(\theta_v(k) + \beta(k)) \\ \theta_v(k) + \Delta \cdot \frac{v(k)}{l_r} \sin \beta(k) \end{bmatrix} \\ \mathbf{u}(k) &= [v(k) \quad \beta(k)]^T \end{aligned}$$

3.6.2 Sensor Model

Given the current vehicle position $\mathbf{x}_v(k)$ and the position of an observed feature $(x_i(k), y_i(k))$, the observations of range, $z_r(k)$, and bearing, $z_\theta(k)$, can be modelled (see Fig. 3.12) as follows [83]:

$$\begin{aligned} \mathbf{z}_i(k) &= \begin{bmatrix} z_{ri}(k) \\ z_{\theta i}(k) \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{(x_v(k) - x_i(k))^2 + (y_v(k) - y_i(k))^2} \\ \arctan\left(\frac{y_v(k) - y_i(k)}{x_v(k) - x_i(k)}\right) - \theta_v(k) \end{bmatrix} + \mathbf{w}_i(k) \end{aligned}$$

where $\mathbf{z}_i(k)$ is the measurement from the i th feature to the vehicle and $\mathbf{w}_i(k)$ is the measurement noise of the i th feature. It can be written in the following general form:

$$\mathbf{z}_i(k) = \mathbf{h}_i(\mathbf{x}_v, \mathbf{p}_i) + \mathbf{w}_i(k) \quad (3.20)$$

where $\mathbf{p}_i = [x_i, y_i]^T$ denotes the i th feature state, $\mathbf{x}_v(k)$ is the vehicle state.

Figure 3.12: The observation relationship between the vehicle and the features in the global coordinate

3.7 Conclusions

In this chapter, the stochastic mapping algorithm by using EKF together with the corresponding map management was firstly introduced. To implement real time algorithms in the later chapters, the hardware and software of our SLAM system have been built. The autonomous vehicle, sensors installed together with their model and the noise characteristics of the laser scanner we used have been analyzed. Real data collection has been performed which shows that the laser sensor noise roughly obeys a Gaussian distribution and a noise variance estimate has also been obtained.

Chapter 4

Feature Extraction Algorithms for SLAM in Outdoor Environments

4.1 Introduction

The term feature is used to denote an object which is “likely” to be detectable when the sensor is moved to new locations. Feature extraction is one of the core problems of SLAM. For outdoor environments, there still lack of systematic methods for feature extraction. This chapter discusses the problem of feature detection for outdoor environments using laser range sensors. In the environments, commonly encountered natural features that can be very useful for mobile robot navigation include edges (large discontinuity), circles or arcs (eg, trees, pillars) and curves. Therefore, we classify features into edges, circles, lines and curves and propose approaches for their extraction.

In [125], Adams presented an edge detection method by EKF. This algorithm can detect edges on line because it processes the data as soon as the range is sensed by the laser sensor. In this approach, lines and edges are considered to be the main features for navigation. However, it does not consider the circular features such as

trees. The use of trees or pillars as features has been investigated in [86, 5]. They addressed the problem of extracting tree trunks from laser scan data where the centre and radius of a circle are estimated by averaging the measurements. This method can be susceptible to outliers which can significantly affect the accuracy of the center and radius estimates.

For more complex outdoor environments, Adams presented another on-line gradient based surface discontinuity detection method for scanning range sensors [6]. In this approach, a range gradient model is derived for outdoor complex surfaces in order to detect large discontinuities in these surfaces. However, the algorithm has high computational complexity and only edges can be detected using this approach.

In this chapter we firstly present two algorithms for feature detection for semi-structured outdoor environments. The first algorithm consists of data segmentation and parameter acquisition. First, a model based data segmentation is applied which divides the collected data into groups that are possibly associated with different features of the environment. EKF or other filtering techniques can be applied for segmentation. Edges are detected during segmentation. We then give a procedure to identify the type of features with which a given group of data is associated. For a circle feature, a modified Gauss-Newton optimization is proposed for fitting circle parameters iteratively. In the second algorithm, features are detected based on model switching and filtering. By establishing appropriate planar and circle models, EKF and UKF based tracking methods are developed where switching between models are carried out based on statistical tests. Compared with [125], the two proposed algorithms can detect not only edges but also circle features by means of on-line and the parameters of the circle features can be obtained accurately.

Further, we note that for some more complex environments, natural features may not be so regular as lines and circles. This requires a more general model to describe the features. We propose another feature extraction algorithm based on a

curve gradient model for this purpose. The curve gradient model is first developed and applied for the data segmentation. The model has the advantage that it is applicable to the features of smooth surfaces such as lines, arcs and circles formed by building walls, trees and lamp poles, etc. The estimated gradients corresponding to a set of segmented data can then be used to identify the type of a feature that the data represents. Feature parameters are obtained after the data segmentation. Compared to the method in [6], our approach can detect more general types of features, such as curves, lines, circle features and edges while the work in [6] is only for edge detection. Another advantage of our approach is the simplification of the system model by defining different system variables, resulting in more efficient computation.

Several experiments are carried out to demonstrate the feasibility and effectiveness of the proposed feature extraction methods. The results show that the proposed algorithm performs very well in outdoor environments.

The layout of the chapter is as follows: Section 4.2 presents a Gaussian-Newton optimization based feature extraction algorithm. Section 4.3 gives a model switching approach to feature detection. Section 4.4 presents a curve gradient model based feature extraction approach. Section 4.5 shows the experimental results using the proposed algorithms in several outdoor environments. Conclusions are drawn in Section 4.6.

4.2 Gauss-Newton Feature Extraction Algorithm

We observe that in many semi-structured outdoor environments, planes such as building walls and cylindrical surfaces such as tree trunks or tree-like objects are often encountered and can be very useful features for mobile robot navigation. Here, we shall firstly propose an algorithm which is able to extract edges and tree trunks with better accuracy compared to [75, 5].

The essential components of our algorithm include two parts: the first is the segmentation of the scan data and the second is the parameter acquisition.

4.2.1 Data Segmentation and Edge Detection

Segmentation is a process of aiming to classify a set of scan data into several groups, each of which possibly associates with different structures of surrounding. The segmentation process is realized through the EKF [125, 81] or other filtering techniques. At each time instant the range estimate is compared to the range measurement based on their statistics in order to decide if an edge has been detected. When the difference between the measured range and the predicted range is beyond a certain threshold, we consider that an edge has been detected. This can be achieved by using a validation gate during the prediction process with the EKF [125, 81].

Planar Model

Let us first introduce a mathematical framework for a planar surface. Consider a vertical plane shown in Figure 4.1 and the corresponding sensed data points from a perfect 3D line of the sight sensor. By applying the sine and cosine rules to the two triangles OAB and OBC in Figure 4.1 and taking into account the fact that the three points A , B and C all lie within the vertical plane, it can be shown that the relationship between successive range readings, when the light beam is incident upon the surface, is given by [125]:

$$d_{i+2} = \frac{d_i d_{i+1}}{2d_i \cos \gamma - d_{i+1}} \quad (4.1)$$

where γ is the constant angle between successive samples of the sensor as it rotates about its vertical axis. Note that the relationship given in equation (4.1) is independent of the elevation angle α .

Figure 4.1: The relationship between successive range readings when scanning a planar surface

System Model

Equation (4.1) is clearly a second order difference equation. Define $x_1(k+1) = d_{i+2}$ and $x_2(k+1) = x_1(k) = d_{i+1}$, where $x_1(k)$ and $x_2(k)$ are the state variables at time instant k . Then (4.1) can be fully defined by the state space equations:

$$\mathbf{x}(k+1) = \mathbf{f}_1(\mathbf{x}(k)) + \mathbf{v}(k) \quad (4.2)$$

where $\mathbf{x}(k+1) = [x_1(k+1) \ x_2(k+1)]^T$,

$$\mathbf{f}_1(\mathbf{x}(k)) = \begin{bmatrix} \frac{x_1(k)x_2(k)}{(2x_2(k)\cos\gamma - x_1(k))} \\ x_1(k) \end{bmatrix}$$

and $\mathbf{v}(k)$ is the process noise which reflects possible imperfection of the surface. We assume that $\mathbf{v}$ is a white noise with variance $\mathbf{Q}(k)$.

Clearly, a small variance $\mathbf{Q}(k)$ implies that the surface is close to be perfect. Equation (4.2) represents a system model which will be used to predict the next

range value from the sensor before the actual range measurement is recorded.

Similar to [125], our observation model is:

$$\mathbf{z}(k) = \mathbf{H}_1 \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \mathbf{w}(k) \quad (4.3)$$

where $\mathbf{H}_1 = [1 \ 0]$ and $\mathbf{w}(k)$ is a zero mean Gaussian noise with a known variance σ_r^2 . The EKF is used to realize the prediction and validation process.

Extended Kalman Filter and Validation Gate

Based on the above system model, an EKF is used to implement the prediction and update. The detail can be found in Appendix B.

In order to identify if a measurement is associated with a new edge (large discontinuity), certain criterion needs to be established. Use the innovation $\nu(k+1)$ and the innovation variance $\mathbf{s}(k+1)$ to define:

$$d(k+1) = \nu^T(k+1)\mathbf{s}^{-1}(k+1)\nu(k+1) \quad (4.4)$$

Note that since ν is a Gaussian random variable, d is a random variable following the χ^2 distribution. The smaller the $d(k+1)$, the higher the probability that the measurement $z(k+1)$ is obtained from the same planar surface. Thus, a validation gate, δ , is used to decide whether the measurement $z(k+1)$ is a close enough match to the predicted data point to continue the filter update. If the measurement is such that $d(k+1) > \delta$, a discontinuity is found. From the χ^2 distribution table, we know that if the observation is from the same planar surface, then $d(k+1) < 6.63$ with a probability of 0.99. If a small δ is selected, there will be more edges found. Here we set $\delta = 6.63$.

After the data segmentation process, we need to decide if each segment of data is associated with a line or a circle (note that the laser sensor data points only form

an arc which is part of a circle, here we call it a circle feature) or a clutter. For a line, the average error between the observation and the EKF prediction at each point should be very small. Note that the prediction error (innovation) sequence $\{\nu\}$ of Equation (B-6) in Appendix B is a Gaussian white noise and its covariance is given by $\mathbf{s}(k)$. Assume that the number of points of the segment is M . Then the sequence $\{\nu\}$ is of the length $M - 2$ (note that the first two points are used to initialize the filter). The average prediction error and its covariance are then given by

$$\bar{\nu} = \frac{\sum_{k=3}^M \nu(k)}{M - 2}, \quad \bar{\mathbf{s}} = \frac{\sum_{k=3}^M (\nu(k) - \bar{\nu})^2}{M - 3} \quad (4.5)$$

Hence, $P\{|\bar{\nu}| \leq 3\sqrt{\bar{\mathbf{s}}}\} = 0.997$. A threshold for the average prediction error can be chosen as $3\sqrt{\bar{\mathbf{s}}}$. The threshold is used to distinguish a line from a circle or a clutter. If the average prediction error is smaller than the threshold, we consider that this segment of data is associated with a line, otherwise, it is associated with a circle or a clutter. To distinguish the circle and clutter, we only need to use the modified Gauss-Newton method that is presented in the following subsection to fit a circle. If the circle fitted can be found in the successive scans, the feature is considered as a circle feature. Otherwise, it should be a clutter.

For a circle, we shall need to estimate its parameters such as the center and the radius of the circle so that future measurements of the circle may be used for robot navigation. In the following, the modified Gauss-Newton method [126] is applied.

4.2.2 Parameter Acquisition

A circle can be defined by the equation $(x - x_0)^2 + (y - y_0)^2 = r^2$ where (x_0, y_0) and r are the center and the radius of the circle, respectively. For a circle fitting problem, a data set of (x, y) is known and the circle parameters (x_0, y_0, r) need

to be estimated. Assume that we have obtained M measurements (x_m, y_m) , $m = 1, 2, \dots, M$, of the circle. Our objective is to find $p = (x_0, y_0, r)$ that minimizes

$$E(p) = E(x_0, y_0, r) = \sum_{m=1}^M [(x_m - x_0)^2 + (y_m - y_0)^2 - r^2]^2 \quad (4.6)$$

This is equivalent to performing the least-squares process using the equations

$$g_m(x_0, y_0, r) = (x_m - x_0)^2 + (y_m - y_0)^2 - r^2 = 0, \quad m = 1, 2, \dots, M \quad (4.7)$$

Equation (4.7) is not linear about the unknown parameters x_0 , y_0 , and r , therefore it is a nonlinear least-squares problem. We propose to use the modified Gauss-Newton optimization method [126] to solve the problem. In our case the Jacobian matrix for the modified Gauss-Newton algorithm is

$$A = \begin{bmatrix} \frac{\partial g_1}{\partial x_0} & \frac{\partial g_1}{\partial y_0} & \frac{\partial g_1}{\partial r} \\ \frac{\partial g_2}{\partial x_0} & \frac{\partial g_2}{\partial y_0} & \frac{\partial g_2}{\partial r} \\ \vdots & \vdots & \vdots \\ \frac{\partial g_M}{\partial x_0} & \frac{\partial g_M}{\partial y_0} & \frac{\partial g_M}{\partial r} \end{bmatrix} \quad (4.8)$$

Let $\bar{g} = (g_1 \ g_2 \ \dots \ g_M)^T$ with g_m as defined in (4.7).

At the k -th step, using the modified Gauss-Newton method to search the solution according to the following equation:

$$(A_k^T A_k + \lambda_k I) \Delta p_k = -A_k^T \bar{g}_k \quad (4.9)$$

where $\Delta p_k = p_{k+1} - p_k$ and p_k is the estimate of $p = [x_0 \ y_0 \ r]^T$ at the k -th iteration. We set the initial value $\lambda_0 = 0.01$ and carry out the following iterations for calculating a suboptimal p :

Step 1: Calculate Δp_k using Equation (4.9);

Step 2: Calculate the sum error $E(p_k + \Delta p_k)$ by Equation (4.6);

Step 3: Compare with the sum error of last step $E(p_k)$. If $E(p_k + \Delta p_k) > E(p_k)$, increase λ_k by a factor of 10, and go back to Step 1;

Step 4: If $E(p_k + \Delta p_k) < E(p_k)$, decrease λ_k by a factor of 10, update the trial solution, i.e. replace p_k by $p_k + \Delta p_k$ and go back to Step 1 until the algorithm converges.

The convergence condition can be defined by the sum of the error square and the number of iterations.

Observe that a starting guess for these parameters is required. We use the first three points (x_i, y_i) $i = 1, 2, 3$ and (4.7) to compute an estimated initial value of (x_0, y_0, r) . The more accurate the initial value is, the faster the algorithm converges.

Remark 4.2 *Note that the Hough transform is commonly used for parameter acquisition [127]. The transform is implemented by quantizing the Hough parameter space into finite accumulator cells. As the algorithm runs, each point is transformed into a discretized curve and the number of intersections of the accumulator cells is counted. However, the problem of how to decide the number of the cells in the parameter space remains unsolved. If the Hough transform is applied for fitting a circle, the parameter space is of three dimensions, which makes the problem more difficult. And with the increased dimension of the parameter space, the Hough transform method becomes more complex and slower. Hence, we use the modified Gauss-Newton method instead of the Hough transform for parameter acquisition.*

Remark 4.3 *Note that the data set is formed after data segmentation, any outlier of measurements has been removed since an outlier produces a large discontinuity in segmentation.*

4.3 Feature Extraction via Model Switching

In the last section, we have presented an off-line feature extraction method. In this section, we shall develop an on-line algorithm, which processes a data point immediately when it arrives, without waiting for the whole scan of data. It should be noted that although the laser sensor outputs a set of scanned data after a full scan, the on-line approach may find applications for future sensors which outputs data one by one. And also the on-line algorithm can be applied to feature extraction with existing sensors which output scan data batch by batch. We observe that some outdoor environment may have features – either plane (such as building walls) or cylindrical surfaces such as tree trunks. Along the path of scanning, it appears that features will switch from one type to another. Therefore, we shall consider a model switching approach in this section. We consider two models (line model and circle model) for semi-structured outdoor environments. Recall that the line model has been given in Section 4.2. In the following we shall establish a circle model and propose a tracking method for this model.

4.3.1 Cylindrical Surface (Circle) Model

As an extension to the on-line planar edge detection model proposed by Adams [81], a cylindrical model for the detection and estimation of possible trees/pillars is now presented. Consider an arc (part of a circle) that is recorded when the 2D laser scans a tree or tree-like object shown in Figure 4.2. When $\alpha = 0$, a geometrical relationship among successive range readings can be found in Figure 4.3 where the laser beam originating from point A intersects the object at the consecutive points A_i , A_{i+1} , and A_{i+2} . The distances between A and these points are d_i , d_{i+1} and d_{i+2} , respectively. Assume first that the circle is perfect and has a center at point O whose coordinate is (x_0, y_0) in a local Cartesian coordinate frame (the origin is at point A) and the radius is r . Thus, OA_i , OA_{i+1} , and OA_{i+2} have the same

length r . It is noted that the reason of assuming $\alpha = 0$ in Figure 4.3 is that this is the usual case when the laser sensor scans the cylindrical.

Figure 4.2: The relationship between successive range readings when scanning a cylindrical surface

System Model

When $\alpha = 0$, by applying the sine and cosine rules to the three triangles OA_iA , $OA_{i+1}A$ and $OA_{i+2}A$ in Figure 4.3, it is not difficult to show that the relationship among successive range readings, when the light beam is incident upon the cylindrical surface, is given by

$$\begin{cases} r^2 = d_{i+1}^2 + a^2 - 2d_{i+1}a \cos(\gamma - \theta) \\ r^2 = d_{i+2}^2 + a^2 - 2d_{i+2}a \cos(2\gamma - \theta) \\ a^2 = x_0^2 + y_0^2 \end{cases} \quad (4.1)$$

Thus, we have:

$$\cos(\gamma - \theta) = \frac{d_{i+1}^2 + a^2 - r^2}{2d_{i+1}a} \quad (4.2)$$

Figure 4.3: The relationship between successive range readings when scanning a cylindrical surface ($\alpha = 0$)

$$d_{i+2} = a \cos(2\gamma - \theta) - \sqrt{a^2 \cos^2(2\gamma - \theta) - a^2 + r^2} \quad (4.3)$$

By substituting (4.1) and (4.2) into (4.3), we obtain that:

$$d_{i+2} = \sqrt{x_0^2 + y_0^2}(\xi - \eta) - \sqrt{(x_0^2 + y_0^2)(\xi - \eta)^2 - x_0^2 - y_0^2 + r^2} \quad (4.4)$$

where

$$\xi = \frac{\cos \gamma (d_{i+1}^2 + x_0^2 + y_0^2 - r^2)}{2d_{i+1}\sqrt{x_0^2 + y_0^2}}$$

$$\eta = \frac{1}{2} \sin \gamma \sqrt{4 - \frac{(d_{i+1}^2 + x_0^2 + y_0^2 - r^2)^2}{d_{i+1}^2(x_0^2 + y_0^2)}}$$

Clearly, d_{i+2} is a function of x_0, y_0, r and d_{i+1} . Let the state variables be:

$$\begin{cases} x_1(k) = x_0 \\ x_2(k) = y_0 \\ x_3(k) = r \\ x_4(k) = d_{i+1} \end{cases} \quad (4.5)$$

Then, we have:

$$\begin{cases} x_1(k+1) = x_1(k) \\ x_2(k+1) = x_2(k) \\ x_3(k+1) = x_3(k) \\ x_4(k+1) = d_{i+2} = g(x_1(k), x_2(k), x_3(k), x_4(k)) \end{cases} \quad (4.6)$$

where $g(x_1(k), x_2(k), x_3(k), x_4(k))$ can be calculated from Equation (4.4). The above can be put into the general form:

$$\mathbf{x}(k+1) = \mathbf{f}_2(\mathbf{x}(k)) + \mathbf{v}(k) \quad (4.7)$$

where $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$,

$$\mathbf{f}_2(\mathbf{x}(k)) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ g(x_1(k), x_2(k), x_3(k), x_4(k)) \end{bmatrix}$$

and $\mathbf{v}(k)$ is a noise vector introduced to reflect the imperfection of the cylindrical surface.

Observation Model

Our observation model is:

$$\mathbf{z}(k) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \mathbf{w}(k) \quad (4.8)$$

$$= \mathbf{H}_2 \mathbf{x}(k) + \mathbf{w}(k) \quad (4.9)$$

where $\mathbf{H}_2$ is the observation matrix, $\mathbf{w}(k)$ is a zero mean additive noise with a known variance σ_r^2 .

The Unscented Kalman Filter (UKF)

From the state Equation (4.7), it is observed that the circle model is highly non-linear. If an EKF is applied, it would result in a very complicated Jacobian matrix and high computational demand in its calculation. Realizing this difficulty, we shall propose to use the UKF [128].

Given the estimate $\hat{\mathbf{x}}(k | k)$ of $\mathbf{x}(k)$ and its estimation error covariance $\mathbf{P}(k | k)$, in order to avoid the linearization involved in the EKF, the UKF works by generating a set of points whose sample mean and sample covariance are $\hat{\mathbf{x}}(k | k)$ and covariance $\mathbf{P}(k | k)$, respectively. The nonlinear function is applied to each of these points in turn to yield a transformed sample, and the predicted mean and covariance are calculated from the transformed samples. The samples are deterministically chosen so that they capture specific information about the Gaussian distribution. For details of the UKF, the readers are referred to Appendix B.

For highly nonlinear systems, the UKF has advantages over the EKF. It avoids the linearization that causes substantial errors in the EKF for nonlinear systems and possible singular points in Jacobian matrices.

Filter Initialization

A reasonable initial estimate of the state and its covariance can be obtained by using the first three measurements. Assuming that the first three points are on

the same circle, we can obtain initial estimates of the state as follows:

$$\begin{cases} \hat{x}_0 = \frac{(y_2 - y_1)(x_3^2 + y_3^2 - x_1^2 - y_1^2) - (y_3 - y_1)(x_2^2 + y_2^2 - x_1^2 - y_1^2)}{2[(x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1)]} \\ \hat{y}_0 = \frac{x_2^2 + y_2^2 - x_1^2 - y_1^2 - 2(x_2 - x_1)\hat{x}_0}{2(y_2 - y_1)} \\ \hat{r}_0 = \sqrt{(\hat{x}_0 - x_1)^2 + (\hat{y}_0 - y_1)^2} \\ \hat{d}_0 = \sqrt{x_3^2 + y_3^2} \end{cases} \quad (4.10)$$

where (x_1, y_1) , (x_2, y_2) , (x_3, y_3) are the coordinates of the first three points which can be obtained from the first three range measurements, d_1 , d_2 and d_3 . Note that the angular position of the laser sensor is assumed to have no uncertainty. Hence, (4.10) can be written as

$$\begin{bmatrix} \hat{x}_0 \\ \hat{y}_0 \\ \hat{r}_0 \\ \hat{d}_0 \end{bmatrix} = \mathbf{g}_0(d_1, d_2, d_3) \quad (4.11)$$

where $\mathbf{g}_0$ is the vector representing the nonlinear functions on the right hand side of (4.10).

The measurements d_1 , d_2 and d_3 can be considered independent. Hence, the initial covariance can be approximated by

$$\mathbf{P}_0 \approx \sigma_r^2 \nabla \mathbf{g}_0 \nabla \mathbf{g}_0^T \quad (4.12)$$

where $\nabla \mathbf{g}_0$ is the Jacobian matrix of (4.11) and σ_r^2 is the variance of the measurement which is known.

4.3.2 Model Switching

In the above sections, we have introduced the line and circle models and their tracking methods. The remaining problem is to determine how to switch between

models in order to detect features of different geometries from a semi-structured environment.

Without loss of generality, assume that we start from a line model whose tracking method has been given in Section 4.2. The EKF can give a predicted distance for each newly arrived point and we can decide if the measurement is close enough to the predicted one by applying the validation gate as explained in Section 4.2.1. If the computed d in Equation (4.4) exceeds the validation gate, the straight line segment being tracked has come to the end and a discontinuity (edge) is found. Then filtering based on a new model (line or circle) is initialized. Note that the filter initialization uses the first three measurements after the discontinuity to calculate the center and the radius of an assumed circle. If the radius is very large, we consider that the next segment is also a line or clutter that is not a circle feature. So we start tracking based on a new line model until the next edge. Note that after the initialization of a line using the first two points, if the following point is not within the validation gate, we shall consider these points a clutter.

On the other hand, if the radius is smaller than a threshold (for example, if only tree trunks are expected, the radius would generally be less than 0.8 meter), we consider that the next point group may represent a circle or a circle shape clutter (points that do not come from a circular shape but can form an arc) and initialize the filtering based on a circle model that is introduced in the last subsection. The process is performed until the next discontinuity. The criterion for discontinuity in circle tracking is the same as that in line tracking using a validation gate.

Remark 4.4 *In our algorithms, since each group of data is formed after data segmentation, any outlier of measurements has been removed since an outlier produces a large discontinuity in segmentation. On the other hand, in complex outdoor environments, features extracted by the above proposed method may become unstable. In order to use these features for navigation, the correspondence between a current feature extracted by the above method and a feature in the map built thus far has*

to be established, which will be discussed in the next chapter.

Remark 4.5 *The two algorithms consider the trees or pillars as the cylindrical surfaces. In some cases, the trees may not be a regular cylinder. That is, the model (4.7) may not be accurate and the algorithms will give rise to errors for the features extracted. But from the experiments of the next section, we shall find that the proposed feature extraction algorithms can perform well in SLAM.*

Remark 4.6 *Compared to the existing method [86, 5], the modified Gauss-Newton algorithm and the model switching algorithm can give more accurate centers of circle features. Compared to the work in [125], they can detect not only edges but also circle features while the approach in [125] is only for edge detection.*

4.4 Curve Gradient Model Based Feature Extraction Approach

In this section, we shall introduce another feature extraction algorithm based on a curve gradient model. It also consists in two parts: the first is the segmentation of data from a scan and the other is the parameter acquisition.

Firstly, we shall establish a general curve gradient model for smooth surfaces. This curve model is shown in Figure 4.4. The laser beam originating from point O intersects the object at the consecutive points A, B, C . The distances between O and these points are denoted as d_i, d_{i+1}, d_{i+2} , respectively. By applying the sine and cosine rules to the three triangles OAB, ABC and OBC in Figure 4.4, it is not difficult to show that the relationship among successive range readings, when

Figure 4.4: A gradient model of a smooth surface for edge detection.

the light beam is incident upon the object surface, is given by

$$\begin{cases} AB^2 = d_i^2 + d_{i+1}^2 - 2d_i d_{i+1} \cos \gamma \\ AC^2 = d_i^2 + d_{i+2}^2 - 2d_i d_{i+2} \cos(2\gamma) \\ AC^2 = AB^2 + BC^2 - 2AC \cdot BC \cos(\pi - \alpha_i) \\ BC^2 = d_{i+1}^2 + d_{i+2}^2 - 2d_{i+1} d_{i+2} \cos \gamma \end{cases} \quad (4.13)$$

Thus, we can obtain:

$$\begin{aligned} d_{i+2} &= \frac{-d_i d_{i+1}^2 \cos 2\alpha_i \cos^2 \gamma + d_i d_{i+1}^2 \cos 2\gamma}{R \cos^2 \alpha_i - (d_{i+1} \cos \gamma - d_i \cos 2\gamma)^2} \\ &\quad \times \frac{d_{i+1} \cos \gamma [-d_{i+1}^2 + (d_i^2 + d_{i+1}^2) \cos^2 \alpha_i - d_i^2 \cos 2\gamma] - d_{i+1} \cos \alpha_i R \sin \alpha_i \sin \gamma}{R \cos^2 \alpha_i - (d_{i+1} \cos \gamma - d_i \cos 2\gamma)^2} \\ &= f(\alpha_i, d_i, d_{i+1}) \end{aligned} \quad (4.14)$$

where

$$R = d_i^2 + d_{i+1}^2 - 2d_i d_{i+1} \cos \gamma$$

and γ is a constant for laser scanners. Since for laser sensors the angle γ is usually small, for a smooth surface we can assume that the change in gradient is constant for the neighboring intersection points, i.e. $\alpha_i \approx \alpha_{i-1}$.

To form a recursive prediction formula, we can rewrite Equation (4.14) by defining:

$$x_1(k) = \alpha_i,$$

$$x_2(k) = d_i,$$

$$x_3(k) = d_{i+1}.$$

Thus, we obtain the smooth surface gradient model of state-space form:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_3(k) \\ f(x_1(k), x_2(k), x_3(k)) \end{bmatrix} + \mathbf{v}_1(k) \quad (4.15)$$

where $f(x_1(k), x_2(k), x_3(k))$ can be calculated from Equation (4.14). By considering possible imperfection of the surface, we incorporate a process noise $\mathbf{v}_1(k)$ in Equation (4.15) and assume that $\mathbf{v}_1(k)$ is a white noise with a given variance $\mathbf{Q}(k)$. Clearly, a small variance $\mathbf{Q}(k)$ implies that the surface is close to be perfect. For a very smooth surface, the $\mathbf{Q}(k)$ can be a very small value, such as 0.0001; For a not very smooth surface, it can be chosen a little bigger.

Similarly to the previous method, our observation model is:

$$z(k) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \mathbf{w}_1(k) \quad (4.16)$$

which can be put into a general form as follows:

$$\mathbf{z}(k) = \mathbf{H}_1 \mathbf{x} + \mathbf{w}_1(k) \quad (4.17)$$

where $\mathbf{w}_1(k)$ is a zero mean Gaussian noise with a known variance σ_r^2 .

The smooth surface gradient model will be used to detect large discontinuities (edges) using EKF as before. If an observed range reading is not within the 3σ bound of the predicted value using the EKF, the filter will be re-initialized and an edge is found.

Remark 4.7 *Compared to the model in [125, 81], our smooth surface gradient model is more general and include the line model for edge detection in [125, 81] as a special case.*

Next, to know the type of the feature each segmented set of data represents, we can use the information of the estimated change of gradient, $\hat{x}_1(k)$, corresponding to the set of data which has been obtained in the segmentation process. For example, if the values of $\hat{x}_1(k)$ are very small for the set of data, we can conclude that the data represents a line feature. On the other hand, if the values of $\hat{x}_1(k)$ are close to a non-zero (not very small) constant, the data may represent an arc or circle. If the values of $\hat{x}_1(k)$ are irregular, the data may come from a random clutter or a feature with a non-smooth surface. Therefore, the proposed smooth surface gradient model based algorithm is a general tool for data segmentation for outdoor environments. It can distinguish the lines, arcs and other shapes of curves.

After the data segmentation, similar methods as in the last sections can be applied to obtain feature parameters. However, it should be noted that the modified Gauss-Newton optimization is used only for those data groups which have more than 3 data points. For the data group which has only 2 to 3 points, we consider it as a point feature whose coordinates can be the average values of the x coordinates and the y coordinates of these points, respectively. For the predominant edges

(large discontinuity—an edge which is far away from other edges), the parameters are obtained during the data segmentation process.

Remark 4.8 *The feature extraction methods proposed in Section 4.2 and 4.3 are mainly for semi-structured outdoor environments where main features are circular features and line features. In unstructured outdoor environments such as rugged terrain, they will fail to work.*

For the curve gradient based feature extraction approach, the failure mode of the algorithm is when it is used in high clustered noisy environments. In this kind of environments, the gradient change of the “curves” often has irregular values. The algorithm cannot judge the type of the “curve” with a variable gradient change.

4.5 Simulation and Experimental Results

4.5.1 Results of Gauss-Newton Feature Extraction and Model Switching Feature Extraction

In this subsection, we shall demonstrate the applicability of the proposed Gauss-Newton and model switching feature extraction algorithms for scanning data from outdoor environments by laser scanner. The laser sensor used in the following experiments is Sick LMS. The field of view is 180 degrees in front of the robot and up to 50 meters of distance. To obtain a 360 degree scene, we use 2 back to back Sick sensors. The range samples are spaced every half a degree, all within the same plane.

In the first experiment, data is collected outdoors as shown in Figure 4.5 where there are 10 pillars labelled from a to j , and the surroundings are building walls and low balusters with small shrubs at a long distance. In this figure, the six cross points represent the six positions at which the robot scans the surroundings.

The laser scanner is placed on the top of a mobile robot at approximately 1.2 meters above the ground. At this level, the sensor can see the objects outside the balusters. In Figure 4.6, we show the real data from one scan of the environment. The origin is the place at which the robot is located. We have done the feature extraction for all the 6 scans. Here we show the feature extraction results for two scans. The results are given in Figure 4.7 to Figure 4.13.

Figure 4.7 to Figure 4.11 show the feature extraction results at position 2 using the Gauss-Newton feature extraction method and the model switching method. Zoomed views of the regions inside the dashed box of Figure 4.7 and Figure 4.10 are given in Figure 4.8 and Figure 4.11, respectively, in order to see the extracted features in the regions clearly. In these figures, the detected edges are denoted by crosses. Similarly, the feature detection results at position 4 are shown from Figure 4.12 to Figure 4.15. It should be noted that the crosses in all the above figures indicate the edges detected in the outdoor environment.

From these figures, we can find that most of the pillars can be extracted by these two algorithms. However, the algorithms sometimes also output circles that don't exist physically, for example, in Figure 4.8, there are 5 circles detected but only 4 pillars (pillar d, e, f, g) exist. The reason for this is that some clutter points form an arc just like true pillar points. This circle does play a similar role in navigation if it is also detected in subsequent scans when the robot moves to other positions. But if these points cannot form a circle when they are observed from another place as the robot moves, then it is regarded as a clutter. In this case, it is useless for the navigation. For example, in Figure 4.8, at position 2, we can extract 5 pillars, but in Figure 4.9, when the laser sensor scans at position 4, there are only 4 pillars detected. In the actual environment, the additional pillar detected at position 2 does not exist. Since it is no longer a circle feature in the subsequent scans, it will not be converted to a "confirmed feature" and will not affect the navigation.

For all the 6 scans, we found that almost all the features can be detected by the

Figure 4.6: Data from One whole scan.

Figure 4.7: Circles and edges extracted from data scanned at position 2 using the proposed Gauss-Newton method (the normal view).

Figure 4.8: A zoomed view of the region inside box A in Figure 4.7.

Figure 4.9: A zoomed view of circles and edges extracted from data scanned at position 4 using Gauss-Newton method.

Figure 4.10: Circles and edges extracted from data scanned at position 2 using the model switching method (the normal view).

Figure 4.11: A zoomed view for the region inside box B in Figure 4.10.

Figure 4.12: The circles and edges extracted from data scanned at position 4 by Gauss-Newton method (the normal view).

Figure 4.13: A zoomed view of the region inside box C in Figure 4.12.

Figure 4.14: Circles and edges extracted from data scanned at position 4 using the model switching approach (the normal view).

Figure 4.15: A zoomed view of the region inside box D in Figure 4.14.

Table 4.1: The summed square errors $E(p)$ of the four circle features using the proposed methods

	pillar d	pillar e	pillar f	pillar g	pillar x
Model switching method	0.0022	0.0007	0.0287	0.0011	—
Modified Gauss-Newton method	0.0010	0.0006	0.0004	0.0006	0.0007

only after a full scan is completed.

To give an indication of the accuracy of the algorithms, we compare our methods with some existing methods reported in literature. In the work by Bailey [5], navigation methods are presented which make use of the circular features from trees. We calculate the relative error of the center of each pillar (CE) and the relative error of the radius of each pillar (RE) separately in terms of the actual and estimated parameters of the circle feature as follows:

$$CE = \frac{\sqrt{(x_{true} - x_{estimate})^2 + (y_{true} - y_{estimate})^2}}{\sqrt{x_{true}^2 + y_{true}^2}} \tag{4.18}$$

$$RE = \frac{|r_{true} - r_{estimate}|}{r_{true}} \tag{4.19}$$

where x_{true} , y_{true} and r_{ture} are the actual coordinates and the actual radius of the circle feature which are obtained from manual measurements and $x_{estimate}$, $y_{estimate}$ and $r_{estimate}$ are their estimated values.

The results are shown in Tables 4.2 and 4.3.

From the tables, we can see that the proposed two new methods are more accurate than the method in [5]. In particular, the Gauss-Newton method gives the most accurate results.

In order to test the feature extraction method for localization, another outdoor experiments has also been carried out for simultaneous localization and map build-

Table 4.2: A comparison of the error CE of the four circle features between the two proposed methods and the method in [5]

	pillar d	pillar e	pillar f	pillar g
Model switching method	0.0192	0.0197	0.0231	0.0141
Modified Gauss-Newton method	0.0179	0.0158	0.0136	0.0137
The method in [5]	0.0251	0.0244	0.0271	0.0166

Table 4.3: A comparison of the error RE of the four circle features between the two proposed methods and the method in [5]

	pillar d	pillar e	pillar f	pillar g
Model switching method	0.0546	0.0542	0.1092	0.0619
Modified Gauss-Newton method	0.0373	0.0011	0.1057	0.0588
The method in [5]	0.1296	0.1326	0.1626	0.1084

ing using the proposed feature extraction algorithms. The experimental environment is shown in Figure 4.16. There are 8 tall trees and building walls and some bushes which constitute the semi-structured outdoor environment. For this semi-structured environment, the main features for localization are tree trunks. The proposed feature extraction algorithms are applied for extracting the features. The vehicle used in the experiment is Cycab, a car-like vehicle, as shown in Figure 3.10. It is equipped with a laser range sensor, Sick LMS, with dead reckoning capabilities. There are four encoders fixed on the wheels of the vehicle. A DGPS with up to 2cm accuracy is used as a reference to give the ground truth of the vehicle pose to get the estimation error.

In the experimental environment, the vehicle moves along the path as shown in Figure 4.17 where the stars denote the trees of the environment, the dashed line indicates the real pose of the vehicle and the solid line means the estimated path using the simultaneous localization and mapping algorithm with the proposed fea-

ture extraction methods. The data association method in the implementation is the same as that in Chapter 5. Figure 4.18 shows a typical laser scanner frame. The dashed line box A indicates the region whose clearer view is shown in Figure 4.19. In these two figures, we can find that there are lines, arcs and edges (point features). However, in the experiment, we only use circle features from the tree trunks for localization. Both the feature detection algorithms can give satisfactory localization results and the accuracy is similar, therefore, we only show the results of the Gaussian-Newton feature extraction algorithm.

To make a comparison on feature extraction performance, we also implement the method in [5]. Figure 4.20 shows the range and bearing innovations of the measurements when we apply the feature extraction method in [5] and our method during the SLAM and their 3σ bounds. The dash-dot line in the middle of each sub-figure is the result of the localization using our proposed feature detection method whereas the solid one in the middle is the result of the localization by the feature detection method in [5]. Figure 4.21 shows the vehicle’s position and orientation errors in the prediction and their 3σ error bounds. Further, we calculate the average absolute estimation error as defined by

$$\Delta x = \frac{\sum |\Delta x_i|}{N}; \quad \Delta y = \frac{\sum |\Delta y_i|}{N}; \quad \Delta \theta = \frac{\sum |\Delta \theta_i|}{N}$$

where Δx_i , Δy_i and $\Delta \theta_i$ are the vehicle pose errors at each time instant and N is the time horizon of the whole localization process. The comparison of the “average absolute error” for the two methods is given in Table 4.4 below.

Table 4.4: The $\Delta x, \Delta y, \Delta \theta$ of the vehicle pose when using different feature detection methods

	Proposed method	Method in [5]
Δx	0.0575	0.0823
Δy	0.0571	0.0732
$\Delta \theta$	0.0353	0.0528

In the table, the unit for Δx and Δy is meter and that for $\Delta \theta$ is radian.

Figure 4.16: The second experimental environment (the whole scene).

Figure 4.17: The estimated path and the true trajectory of the vehicle during the SLAM.

Figure 4.18: A whole scan data of the second experiment environment corresponding to Figure 4.16.

Figure 4.19: The circle features (trees) and edges extracted from the environment in Figure 4.16 using Gauss-Newton algorithm.

Figure 4.20: A comparison on range and bearing innovations during the localization when using the proposed method (dash-dot line) and the method in [5] (solid line in the middle).

Figure 4.21: The error and 3σ error bounds of the vehicle when using different feature detection methods. The dashed line is the result of the localization using the proposed feature detection methods.

4.5.2 Results of Curve Gradient Model Based Feature Extraction

Simulation Results

To test the curve gradient model based feature extraction method, we create a simple artificial environment. There are 3 trees, a wall and a curve surface. They have different gradient changes when we track the surface points by using EKF. The simulated scan data for one scan and the features extracted from it can be seen in Figure 4.22. In the simulation, we first generated the data for the features without noise and then added Gaussian noise with the zero mean and the same variance. Monte carlo runs (50 times) with random noise are performed. We calculate the average relative errors of the extracted features and compared the results with the method in [6].

The relative error of the center of each tree in terms of the actual and estimated

Figure 4.22: The scan data of the simulation environment and the feature extraction results

parameters of the circle feature is as follows:

$$\frac{\sqrt{(x_{true} - x_{estimate})^2 + (y_{true} - y_{estimate})^2}}{\sqrt{x_{true}^2 + y_{true}^2}} \quad (4.20)$$

where x_{true} and y_{ture} are the actual coordinates of the center of the circle feature and $x_{estimate}$ and $y_{estimate}$ are their estimated values.

Similarly, the relative errors of the extracted edges are also calculated according to Equation (4.20).

In the simulation, the curve gradient model based feature extraction algorithm is performed with data segmentation and parameter acquisition. After the edges are extracted and the types of features are distinguished according to the curve gradient change, the features' parameters are obtained. For the circle features, it is obtained by the modified Gauss-Newton algorithm.

We also compare the calculation time of the two algorithms by simulations on the same computer.

The results are shown in the table below.

Table 4.5: A comparison of the relative error of the circle features and edge features between the proposed method and the method in [6]

	tree 1	tree 2	tree 3	edge 1	edge 2	edge 3	CPU time(s)
Curve gradient model based method	0.0105	0.0082	0.0165	0.0108	0.0221	0.00689	0.090
The method in [6]	–	–	–	0.0199	0.0204	0.00679	0.153

From the above table, we can see that the proposed curve gradient model based approach can detect both the circles and edges including the edges of the curves whereas the method of [6] can only detect the edges. Both the methods have a similar accuracy for edges. However, our proposed method is computationally more efficient due to the use of a simpler state space model.

Remark 4.9 *When an environment has many pillars or trees, our method will perform better than Adams’ method which cannot detect circle features. For the environments where edges are significant, the proposed method can give similar results as the method in [6] with less computational requirement.*

Experimental Results

The navigation loop is based on encoders and range/ bearing information provided by a laser sensor. The models for the process and observation are highly nonlinear. The encoders provide velocity and steering angle information that is used with a kinematic model of the vehicle to predict position and orientation. The prediction is updated with external range and bearing information provided by a laser sensor. A GPS is also used as a reference to judge if the pose estimation of the vehicle is accurate or not.

This implementation is based on particle filter. Because the problem concerned here is feature extraction, the details about particle filter will not be introduced here but can be found in Appendix C. We use the Nearest Neighbor data association method here. The squared normalized innovation between the observation and the feature location, d_i^2 , is compared against a validation gate for the association being considered.

$$d_{i,k}^2 = \hat{\nu}_{i,k}^T \hat{s}_{i,k}^{-1} \hat{\nu}_{i,k}$$

where $\hat{\nu}_{i,k}$ and $\hat{s}_{i,k}$ can be calculated from Equations (C-11) and (C-12) in Appendix C for each feature. The validated measurement nearest to the predicted measurement is used for updating the state of the vehicle.

The curve gradient model based feature detection algorithm is tested in two different real outdoor environments.

Experiment 1

In order to test the robustness of the proposed curve gradient model based feature detection algorithm, a dynamic outdoor environment where a bus is passing by is firstly explored by our vehicle. In order to improve our map accuracy, we apply two type features—point feature and circle feature as described in Section 4.4.

In this environment, the features are more complex and irregular than those in the last subsection. Figure 4.23 shows the real campus map including the road of the experimental trajectory. It is a line path which is between Canteen 3 and Block S2 at Nanyang Technological University. The vehicle starts from point A and stops at point B. There are several trees and some building walls on both sides of the road. Figure 4.24 to Figure 4.27 give a set of typical laser scans during which a moving bus was passing by our vehicle. From scan 1038 to scan 1076, we can see that the speed of the bus was very fast because the features of the environment did not change greatly while the bus's position changed greatly during the short

Figure 4.23: Map of the experimental environment.

period. For this environment, the main features that were extracted from the scan data include point feature and circle features. The features extracted can be seen in Figure 4.28 (For circle features, we only plot their centers in this figure). The features used here are only on one side of the road in order to reduce the computational requirement. We can see that for the dynamic environment, the feature detection method can perform well. The experimental localization results can be seen in Figure 4.28 where the estimated path almost overlaps with the GPS reading of the path and the error is no more than 0.5 meters. Figure 4.29 and Figure 4.30 give the range and bearing innovations during the localization process. It should be noted that in the experiment we only use features on one side of the road rather than those on both sides of the road in order to reduce the computational cost during the localization process.

Figure 4.24: Scan1038

Figure 4.25: Scan1056

Figure 4.26: Scan1066

Figure 4.27: Scan1076

Figure 4.28: The features and the comparison of the truth path and the estimated trajectory in the dynamic environment.

Figure 4.29: The range observation innovation during the localization process.

Figure 4.30: The bearing observation innovation during the localization process.

Experiment 2

The other testing site is a long walk way around Hall 7 at Nanyang Technological University. The real campus map including this road is shown in Figure 4.31. Dead reckoning sensors and laser range sensor are combined to predict the vehicle's trajectory using the EKF and to build the map at the same time. During the SLAM, in order to improve our map accuracy, we apply two types of features, namely point features (edges) and circle features, detected by the proposed method and the data association algorithm in Chapter 5.

For circles features, the proposed modified Gauss-Newton optimization method is applied to obtain their parameters. For a data group which has only 2 to 3 points, the average values of the x coordinate and the y coordinate are taken to be the parameters of the point features. For predominant edge features, the parameters have been obtained during the data segmentation process. It should be noted that here a predominant edge feature means an edge which is far away from other edges. In our experiment, we set the distance to be 5 meters.

Figure 4.32 gives a typical laser scan during the process. Figure 4.33 shows the circle features extracted from a laser scan for the region inside Block E of Figure 4.32. Since the environment involves a large number of edges, we only use those predominant edges of the road. During the SLAM, in order to reduce the number of noisy edges, we adopt the strategy—if an edge feature is not observed in 10 successive scans, we consider it out-of-dated and remove it from the map. Tentative features that are not re-observed are also removed from the map after a fixed time interval has elapsed.

Figure 4.34 shows the experiment results of the SLAM. We can see that the estimated path is very close to the GPS reading. In this figure, there is a break of the GPS data (the GPS data coordinate is vacant roughly between $(-10, -120)$ and $(-30, -115)$). That is, no ground truth is available in this segment of road.

Figure 4.31: The map of our experiment site.

This is because the vehicle is under a building to make a turn where the GPS signal is blocked. The average position error of the entire SLAM process except the segment where the ground truth is unavailable is smaller than 0.5 meter.

4.5.3 Discussions on the Proposed Algorithms

All of the proposed feature extraction algorithms in this chapter can detect edge features for outdoor environments by an on-line manner.

The modified Gauss-Newton algorithm and the model switching algorithm can give more accurate centers of circle features compared to the existing method [86, 5] and [125].

The curve gradient model based algorithm can detect not only edges but also other

Figure 4.32: A typical laser scan during the vehicle moving.

types of features including circle features whereas the closely related algorithm in [6] can only extract edges. It is also worth mentioning that our curve gradient model is simpler than that of [6].

4.6 Conclusions

For semi-structured outdoor environments, two new algorithms for feature detection have been firstly presented in this chapter. They can be used for extraction of planar surfaces, tree trunks or tree-like objects and edges in semi-structured outdoor environments for mobile robot navigation. Experiment results show that both the methods can effectively extract features for navigation purpose. They, however, also output features which do not physically exist. In order to reduce the instability of the extracted features, data association is to be introduced in the

Figure 4.33: The features (circle features and edges) extracted from the zoomed view of the block E in Figure 4.32.

Figure 4.34: The map and the estimated path and GPS reading of the path during the SLAM. In the figure, the third feature (from the start point of the trajectory) is at (17.8,-26.7) and its covariance is $\sigma_{xx} = 0.1748$; $\sigma_{yy} = 0.0955$; $\sigma_{xy} = \sigma_{yx} = 0.0146$

following chapter.

For more complex environments, a curve gradient model based feature extraction approach was developed. Like the Gauss-Newton feature extraction, it also consists in two steps: data segmentation and parameter acquisition. But the model for data segmentation is different. The curve gradient model can be applied not only for data segmentation but also for determining various types of features (such as lines, arcs on a circle or other curves) according to the estimates of the gradient changes. The feature detection algorithms have been realized in real outdoor environments and satisfactory results have been obtained.

Chapter 5

An Efficient Data Association Approach for SLAM

5.1 Introduction

A feature based approach to SLAM is to use the information obtained by sensors mounted on a vehicle to build and update a map of the environment and compute the vehicle location in that map. One of the critical problems in obtaining a robust SLAM solution is data association, i.e. relating sensor measurements to features in the map that has been built thus far [129]. SLAM relies on correct correspondence between data obtained from the robot sensors and the data currently stored in the map.

Data association based on 0-1 integer programming problem for multi-sensor multi-target tracking was firstly proposed in [98] and later improved in [99, 130, 131]. In these works, the data association and tracking problem was formulated as a multi-dimensional assignment problem.

By extending the multi-sensor and multi-target tracking problem, the data associ-

ation in SLAM is formulated as a 0-1 integer programming problem in [2, 3, 132]. In [2, 3], the data association in SLAM is first formulated as a three dimensional assignment problem where two frame scan data is applied to establish the correspondence between the measurements and the features stored in SLAM. This algorithm can track features very reliably especially in high clutter density environments. However, three dimensional assignment problem is NP-hard [135] and only heuristic optimization algorithms are available for an approximation solution because of the computational complexity. In [132], a parallel development of data association in SLAM is formulated as a two dimensional assignment problem. The chapter reports the detail of this work.

In this chapter we present an efficient integer programming (IP) based data association approach to SLAM. In this approach, the feature based SLAM data association problem is formulated as a 0-1 IP problem that considers only 1 frame scan data. Therefore, it is formulated as a two dimensional assignment problem for which many optimization algorithms such as those in [130, 131, 133, 100, 134] can be applied. The IP problem is approached by first solving a relaxed linear programming (LP) problem. In order to reduce the computational burden, a validation gate is applied to reduce the size of the solution space. An iterative heuristic greedy rounding (IHGR) process based on linear programming techniques [133, 100, 134] is then proposed to obtain a suboptimal solution to the integer programming problem. The algorithm has moderate computational requirements.

Detailed simulation and experimental results show that the proposed method gives a much higher success rate of data association for environments of high density features than the NN algorithm while the cost of computation is moderately higher than the latter. Further, the experiment in a real outdoor environment shows that the NN algorithm leads to a diverged vehicle pose estimate whereas the proposed algorithm performs satisfactorily. As compared to other existing methods such as the JCBB algorithm, our approach has a lower computational complexity and provides a good trade-off between accuracy and computational cost. We also com-

pared the algorithm with the most closely related work in [2, 3] with simulations. The simulation results show that a faster calculation speed can be achieved while compromising the rate of successful correspondences.

The chapter is organized as follows: Section 5.2 is devoted to an IP formulation for data association in SLAM. Section 5.3 presents an iterative heuristic greedy rounding algorithm for the IP problem. Section 5.4 shows some simulation and experimental results. Some conclusions are drawn in Section 5.5.

5.2 Problem Formulation

In this section we formulate the data association of SLAM as a 0-1 integer programming problem similar to [2, 3]. A mathematical framework of SLAM which is based on the EKF will be applied.

Data association of SLAM is a decision process of associating measurements (observations) with existing features in the stochastic map. It should be noted that the term “measurements” (observations) in this chapter refers to the observed features after feature extraction rather than the raw sensor measurements. Generally, the number of the measurements obtained in each scan is not equal to the number of features whose positions are estimated by the EKF. Each measurement may either (1) belong to a previously known geometric feature or (2) be a new geometric feature or (3) be a spurious measurement (also called a false alarm). On the other hand, there also exist features that do not have associated measurements in the current scan. A dummy element is applied to denote the case of a false alarm or a new feature or a feature that does not have an associated measurement here.

Assume that there are M measurements from the latest scan which are to be assigned to N existing features in the map built based on the previous scans. It is also assumed that the measurements are independent. Typically, $M \neq N$. Define

the binary assignment variable

$$x_{nm} = \begin{cases} 1 & \text{if measurement } m \text{ is assigned to feature } n \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

Note that $x_{n0} = 1$ implies that the feature n has no associated measurement in the current scan, and $x_{0m} = 1$ implies that measurement m is not assigned to any of the existing N features, but instead, assigned to a dummy feature—false alarm or newly initialized feature. In the data association process, we make the reasonable assumption that one measurement originates from at most one feature, and one feature can produce at most one measurement. Therefore, the following constraints can be imposed to the association variables:

$$\sum_{m=0}^M x_{nm} = 1, \quad n = 1, 2, \dots, N \quad (5.2)$$

$$\sum_{n=0}^N x_{nm} = 1, \quad m = 1, 2, \dots, M \quad (5.3)$$

Our goal is to match the sensor's observations with the features by providing estimates of features' positions relative to the vehicle pose at the time of the current scan. In order to formulate the 2-D assignment problem, a generalized likelihood ratio which involves feature state estimates for the candidate associations is used to assign a cost to each association. Similarly to the multi-target tracking problem [131], we maximize a likelihood function LH as follows:

$$LH = \prod_{\{n,m \in E_{nm}\}} \Lambda(z_m, f_n) \quad (5.4)$$

$$\Lambda(z_m, f_n) = \frac{1}{2\pi|\mathbf{S}|^{1/2}} \exp\left\{-\frac{1}{2}[z_m - \hat{z}_n]^T \mathbf{S}^{-1}[z_m - \hat{z}_n]\right\} \\ m \neq 0, n \neq 0 \quad (5.5)$$

where z_m is the m -th measurement of the scan, $\hat{z}_n$ is the predicted relative position of the n -th feature by the EKF, $\mathbf{S}$ is the covariance matrix of $z_m - \hat{z}_n$, and E_{nm} is the set of all possible assignment pairs. The likelihood ratio $\Lambda(z_m, f_n)$ is in fact the probability that the m th measurement matches the n th feature in the current sensor scan. In order to constitute a 2D-assignment optimization problem, instead of maximizing the product of matching probabilities, we can minimize the negative log-likelihood ratio. To this end, define:

$$C_{nm} = -\ln \Lambda(z_m, f_n) \quad m \neq 0, n \neq 0 \quad (5.6)$$

Then, an equivalent cost function for (5.4) can be written as follows:

$$\min \sum_{\{n, m \in E_{nm}\}} C_{nm} x_{nm}$$

where $C_{nm} = 0$, when $m = 0$ or $n = 0$. Thus, the data association of SLAM can be formulated as the following 0-1 integer programming problem:

$$\min \sum_{\{n, m \in E_{nm}\}} C_{nm} x_{nm} \quad (5.7)$$

subject to

$$\sum_{m=0}^M x_{nm} = 1, \quad n = 1, 2, \dots, N \quad (5.8)$$

$$\sum_{n=0}^N x_{nm} = 1, \quad m = 1, 2, \dots, M \quad (5.9)$$

where $x_{nm} \in \{0, 1\}$ and

$$C_{nm} = \begin{cases} 0 & \text{if } m = 0 \text{ or } n = 0 \\ -\ln \Lambda(z_m, f_n) & \text{otherwise} \end{cases} \quad (5.10)$$

In this algorithm, two points should be noted:

- if a measurement does not fall into the 3σ region of any of the existing N features (see the gating process in Subsection 5.3.1), we assume that it is a new feature and add it to the SLAM map directly. It will no longer be considered in the above 2D-assignment.
- if an existing feature does not have any measurement that falls into its 3σ region, we consider that this feature is not observed in the current scan. No further matching will be carried out for this feature.

5.3 The IHGR Based Data Association Algorithm

In this section, we apply an LP based algorithm to solve the data association problem formulated in the last section. The method is a combined IHGR and LP algorithm. In order to reduce the computational burden, a validation gate is applied first to reduce the above global association to several local associations.

5.3.1 Gating

In order to reduce the solution space, gating is first applied. Only measurements that are close enough to the predicted state of an existing feature are considered possible candidates of association with the feature. The criterion of gating is given by:

$$\tau_{ij} = \nu_{ij}^T \mathbf{S}_{ij}^{-1} \nu_{ij} \leq \varepsilon$$

$$i = 1, 2, \dots, N; \quad j = 1, 2, \dots, M$$

where

$$\nu_{ij}(k+1) = \mathbf{z}_j(k+1) - \hat{\mathbf{z}}_i(k+1 | k)$$

and $\mathbf{S}_{ij}$ is the covariance of the innovation ν_{ij} . In this equation, $\hat{\mathbf{z}}_i(k+1 | k)$ can be calculated from Equation (3.6) and $\mathbf{z}_j(k+1)$ means the j th measurement in the current scan.

Note that since ν_{ij} is a Gaussian random variable, τ_{ij} is a random variable following the χ^2 distribution. Thus, a validation gate, ε , is used to decide whether the measurement $\mathbf{z}_j(k+1)$ is a close enough match to the predicted feature position. From the χ^2 distribution table, we know that $\tau_{ij} < 6.63$ with a probability of 0.99 for a variable of two degrees of freedom. Here we set $\varepsilon = 6.63$.

5.3.2 Iterative Heuristic Greedy Rounding

Relaxation and Rounding Technique

Many combinatorial optimization problems can be attacked with approximation algorithms that yield a feasible solution in polynomial time with cost close enough to the optimal one [135]. Such approximation algorithms can be loosely categorized as combinatorial approximation algorithms and LP based approximation algorithms. Here, we are interested in the latter category. In order to solve the IP problem, we often firstly relax it into a LP problem [136]. In general, the relaxation refers to the action of relaxing the integer requirement of a linear IP to turn it into an LP. However, the optimal solution to the LP problem in general does not coincide with the solution to the initial IP problem. One of the basic techniques which is widely exploited to derive a LP based approximation algorithm is LP-based rounding. It refers to how to construct a feasible solution for the IP from the LP [137, 136]. Thus, the LP-based approximation algorithm first relaxes the IP problem to a LP problem, solves the LP problem and then converts the fractional optimal solution of LP to an integer solution. The heuristic algorithm applied here is based on the relaxation and rounding algorithms as described in [133, 100, 134]. These algorithms are used for other applications than data associ-

ation in SLAM. It is noted that the two dimensional assignment problem can also be solved by other optimization techniques other than those in [133, 100, 134].

IHGR Procedure

By changing the integer constraint $x_{nm} \in \{0, 1\}$ to $0 \leq x_{nm} \leq 1$, the IP problem is relaxed to a LP one. The LP problem can be solved by basic LP algorithms, such as the Simplex algorithm [138]. If the optimal solution x_{op} of the LP-relaxation is fully integer-valued (in this case all decision variables will have the value of either 0 or 1) then the solution x_{op} is optimal for the 0-1 IP problem in Equation (5.7) [137]. Otherwise, we apply the IHGR procedure (see, e.g. [133]). Observe that the larger the decision variable x_{nm} , the higher the probability that the m -th measurement associates with the n -th feature. Hence, the algorithm starts by setting the maximum decision variable (with a value close to 1) to 1 and all other entries in the same row and column to zero to meet the constraints (5.8) and (5.9). Then, solve the LP problem for the remaining assignment matrix and repeat the IHGR procedure to decide the next pairing of measurements and features. The process is continued until all measurements have been assigned. In this manner, a feasible (but not necessarily optimal) solution for the original IP problem is constructed.

In the IHGR procedure, when x_{nm} is set to 1, all variables in the column and row associated with the specific set in E_{nm} must be set to 0. Once a variable is forced to a certain value, it is not allowed to change any more. To achieve this, all rounded variables and all implicated variables are discarded from the IHGR procedure. In this way, the IHGR will never set the value of a variable twice. This deletion of variables also applies to the initial LP solution, i.e. all variables with value 1 and all zero-valued variables implicated by them, are removed. The IHGR algorithm repeats the actions of selection, rounding and deletion until there are no variables left. The outcome will then be a feasible solution to (5.7).

The algorithm described above can be summarized in the following 4 steps:

- Step 1: Relax the IP problem to a LP problem and solve the LP problem. If the solution is fully integer valued, then stop. Otherwise, go to Step 2.
- Step 2: Set the maximum decision variable to 1 and other entries in the same row and column to zero.
- Step 3: Delete the elements that have been decided, go to step 1 for the rest of the assignment matrix.
- Step 4: Repeat the above relaxation, selection, rounding, and deletion steps until all the elements are assigned.

Observe that the IHGR can be implemented efficiently, but it is clear that there is no guarantee that this heuristic procedure yields the optimal solution to the IP problem. However, the simulations and experiments to be discussed in the following section show that the IHGR does result in acceptable feature-measurement assignments of which the achieved cost is close to the optimal cost.

5.3.3 Algorithm Complexity

Due to the application of the gating process that is affected by random factors, we cannot give an exact description of the complexity. However, we know that in any fixed dimension, LP can be solved in strongly polynomial linear time (linear in the input size) [139]. For our case, the input size is $M \times N$. Thus, we can roughly know the worst-case complexity of the proposed algorithm is $O(MN + (M - 1)(N - 1) + \dots + (|M - N| + 1) \times 1)$.

Neira and Tardós [60] presented a data association approach—JCBB. JCBB performs incremental construction and search of an interpretation tree of joint association hypotheses. The gating determines acceptable hypotheses and performs branch and bound pruning of the search space. The discussion in [60] does not

provide any theoretical bound, but gives an empirical complexity estimate of $O(1.53^N)$, where N is the number of observed features. When N is large, the algorithm will have a high computational complexity. For example, when $N = 30$, $1.53^{30} = 347330$. But for our algorithm, $MN + (M - 1)(N - 1) + \dots + (|M - N| + 1) \times 1 = 9455$ where $M = 30$ that is the worst case. Therefore, when the observed feature number is large (such as more than 30), our algorithm can work much more efficiently than the JCBB. In the simulation, we also found that when N is large, for example $N \geq 30$, JCBB is too slow to work while the algorithm proposed can work well.

5.4 Simulation and Experimental Results

The algorithm presented was tested in two different environments, an artificial environment and a real outdoor environment. In these two environments, SLAM was implemented by using the data association algorithm proposed in the last section.

5.4.1 Results of Simulation Environment

Simulation Example 1

The first test environment is established by randomly generating some features and assuming that the vehicle's trajectory is a circle whose radius is 62 meters. The robot moves at a constant speed and the heading angle changes 1 degree at each sampling instant. The environments involve 105 features which are randomly distributed in a region of 120 meters by 120 meters. The vehicle model used in the simulations is the same as that in [124]. The standard deviations of the laser sensor are $\sigma_r = 0.01$ meter and $\sigma_\theta = 0.0005$ radian. When the vehicle moves, some of the features are observed. We assume that the features' positions are unknown

which is the case in SLAM and match the features with the observations by using the NN and the IHGR algorithms, respectively.

In order to compare the successful rates of these two data association algorithms, we change feature positions except one feature 200 times to implement the SLAM. The features' positions are randomly generated each time with uniform distribution in space. We apply the NN data association algorithm and the IHGR data association method to perform the SLAM processes, respectively. In the simulation, we only fix the position of one feature and investigate the successful rate of the matching of this feature with its observation in one scan (scan 15) during the SLAM. The data association successful rate for this feature when using IHGR algorithm is 96.5% (193/200) while it is only 81% (162/200) for the NN algorithm.

Figure 5.1: The unsuccessful mapping when applying the NN data association algorithm in the SLAM process.

When the features are closely located, the NN algorithm fails. Figures 5.1 and 5.2 show the SLAM results where the NN algorithm fails whereas the proposed IHGR algorithm performs well. In this case, the positions of 3 features in the environment

are fixed and they are located at (27, 20.5), (26, 19.5) and (26.5, 19), respectively. The remaining features are randomly distributed. It can be observed from Figure 5.1 that the NN algorithm leads to diverged estimates of vehicle pose and feature positions. On the other hand, our IHGR method performs very well as observed in Figure 5.2. In fact, the vehicle's true path is almost overlapped with the estimated one during the SLAM process. The global error between the estimated vehicle's position and the ground truth can be seen in Figure 5.3.

A comparison on execution time between the NN algorithm and the IHGR based data association algorithm versus the number of features observed in one scan is shown in Figure 5.4 (the algorithms are run on Pentium IV PC, 1.7GHz) for the cases when the NN algorithm is able to give successful SLAMs. In the figure, the mean execution time means the average CPU time used for 50 Monte Carlo runs for data association algorithm. In the simulation, we extract data at each scan under different feature number and assume that the number of measurements is the same as that of the existing features (this is the most time consuming case). The result shows that our IHGR based method has moderate computational requirement and is implementable in real-time applications.

In Table 5.1, we show that the average global errors of the vehicle's position estimate versus the number of features observed (observations) in each scan in the same environment mentioned above. We fixed the number of observations in each SLAM process which lasted for a few thousand steps but changed the number of observations for different SLAM processes. The standard deviations of the laser sensor are $\sigma_r = 0.01$ meter and $\sigma_\theta = 0.0005$ radian. In the table, the errors are calculated as follows:

$$\text{X direction error} = \frac{1}{K} \sum_{i=1}^K |E_{x_i}|, \quad (5.11)$$

$$\text{Y direction error} = \frac{1}{K} \sum_{i=1}^K |E_{y_i}| \quad (5.12)$$

Figure 5.2: The mapping and vehicle path when applying IHGR data association method in the SLAM process.

Figure 5.3: The vehicle's position errors in the global coordinate between the ground truth (generated data) and the estimated position using IHGR data association algorithm in the SLAM process.

Figure 5.4: The mean execution time of the IHGR algorithm and the NN algorithm. The mean execution time means the average CPU time used of 50 Monte Carlo runs for the process of the data association algorithm. The mean execution time of IHGR is nearly linear with respect to the observed feature number by repeated experiments.

Table 5.1: The comparison of the SLAM performance for various number of observations (M).

	Errors(Meters)	
M	X direction error	Y direction error
2	0.2374	0.2518
5	0.1737	0.1946
8	0.1278	0.1469
10	0.1008	0.1348
13	0.0792	0.0986
15	0.0771	0.0910
18	0.0631	0.0874
20	0.0558	0.0806
25	0.0501	0.0645
30	0.0354	0.0439
35	0.0323	0.0357

where $|E_{x_i}|$ and $|E_{y_i}|$ are the absolute values of the i th time step vehicle position errors in X direction and Y direction, respectively, and K is the total number of steps during the SLAM.

It is found that the errors decrease when the number of observations increases. When the number of observations is more than 35, we found that the errors change very little. Therefore, we only show the results when the number of observations is smaller than 35.

In order to examine the robustness of the IHGR based algorithm with respect to sensor noise, we also carried out the simulations under various sensor range variances. From [125], we know that for any range finder, the range noise covariance can vary depending on the received signal amplitude while the angular uncertainty is relatively very small and little changed compared to the range noise. In the simulation, we fix $\sigma_\theta = 0.0005$ radian. The sensor used here is a LADAR (Laser Detection and Ranging) sensor, therefore, the standard deviation of the range noise σ_r can be typically from 0.01 meter to 0.25 meter [125] for different sensors. Table 5.2 shows the average performance of 10 SLAM processes for the same environments mentioned earlier with 12 observations.

In Subsection 5.3.3 we have compared the computational burden of our method with that of the JCBB algorithm [60]. Here we shall compare the accuracy of these two methods. Note that the JCBB algorithm which uses the branch and bound enumeration algorithm gives an optimal solution to the data association problem. In this simulation, we use 30 features in the large scale map because the speed of the JCBB algorithm becomes quite slow when the feature number is too large.

From Table 5.3 one can see that with the IHGR based data association method a near optimal solution to the SLAM has been achieved. Observe that when the number of features is 5, the same performance as the JCBB algorithm which gives the optimal solution is obtained in this experiment.

Table 5.2: SLAM performance versus the standard deviation of the sensor range noise.

	Errors(Meters)	
σ_r	X direction error	Y direction error
0.01	0.0815	0.0979
0.02	0.0989	0.1135
0.03	0.1247	0.1389
0.05	0.2049	0.2209
0.08	0.2975	0.3108
0.1	0.3564	0.4532
0.15	0.4753	0.6786
0.2	0.6785	0.9860
0.25	0.9076	1.0062

Table 5.3: The comparison of the SLAM performance for the JCBB algorithm and IHGR algorithm under different number of features. The unit for the errors is meter.

	JCBB method		IHGR method	
M	$Error_x$	$Error_y$	$Error_x$	$Error_y$
5	0.1802	0.1899	0.1802	0.1899
8	0.1346	0.1573	0.1389	0.1623
10	0.1025	0.1389	0.1101	0.1475
13	0.0834	0.1003	0.0924	0.1120
15	0.0799	0.0915	0.0856	0.1079
18	0.0702	0.0832	0.0786	0.0886
20	0.0628	0.0784	0.0703	0.0826
25	0.0591	0.0748	0.0651	0.0799
30	0.0528	0.0537	0.0580	0.0722

Table 5.4: The typical scan matching results.

	Existing feature number and dummy element							
		1	2	3	4	5	6	0
Measurements	1	0	0	1	0	0	0	0
	2	0	0	0	0	1	0	0
	3	0	0	0	0	0	1	0
	0	1	1	0	1	0	0	0

Simulation Example 2

We also designed another set of simulations that are similar to the one in [2, 3] to check the robustness of the IHGR data association and computational cost in comparison with other data association approaches. In the simulations, the vehicle follows the same path as that in simulation example 1. The vehicle and sensor models used are the same as the above simulations. The feature number is 20 and the size of the environment is changed between $50 \times 50\text{ m}^2$ and $40 \times 40\text{ m}^2$ in these simulations. The features are uniformly distributed in the SLAM area. The clutter is also uniformly distributed with density of $0.001/\text{m}^2$ and $0.005/\text{m}^2$. Table 5.4 showed the matching results of a typical scan in the SLAM process. The hypothesis number is 6 and the false alarm number is 1.

The IHGR data association was also compared with the NN algorithm, JCBB algorithm and two frame assignment data association in [2, 3] by performing Monte Carlo runs. For simplicity, the two frame assignment data association is denoted as the TFDA algorithm in the following tables. In the first simulation, 200 Monte Carlo Runs are used for each data association. We investigate tracking loss rate of two features. Here we define that if the correct measurement is not in the 99% validation region of the estimation of the feature measurement for more than 3 scans, the feature’s tracking is considered to be lost. The simulation results can be seen in Table 5.5. In the simulation, we also calculated the corresponding average CPU time used for one cycle data association of the two features. The results are

Table 5.5: The comparison of the features’ tracking loss rate of several data association methods.

	Tracking loss rate of false measurement density of 0.001 (%)		Tracking loss rate of false measurement density of 0.005 (%)	
	<i>Feature1</i>	<i>Feature2</i>	<i>Feature1</i>	<i>Feature2</i>
NN	6.5	5	13.5	12
IHGR	4	3.5	7.5	7
TFDA	3.5	3	5.5	4.5
JCBB	2	1.5	3	2.5

Table 5.6: The comparison of the computational time of several data association methods.

	Average CPU time under false measurement density of 0.001	Average CPU time under false measurement density of 0.005
NN	0.0093	0.0134
IHGR	0.0165	0.0821
TFDA	0.0576	0.1755
JCBB	0.1069	0.3752

shown in Table 5.6.

In the second simulation, we fixed the false measurement density of 0.001, but changed the feature density from 0.008 to 0.125 to check the features’ tracking loss rate and the CPU time used. Here the feature density is defined by $D = \frac{F}{A}$, where F is the number of features in the simulation environments and A is the area of the space. The results can be seen in Tables 5.7 and 5.8, respectively.

From the tables, we can see that the IHGR method can perform better than the NN method when the feature density and false measurement density is high. Compared to the TFDA and JCBB algorithms, it is computationally more efficient but with slightly higher tracking loss rates when the false alarms and feature density are

Table 5.7: The comparison of the features’ tracking loss rate of several data association methods under different feature densities.

	Tracking loss rate of feature density of 0.008 (%)		Tracking loss rate of feature density of 0.0125 (%)	
	<i>Feature1</i>	<i>Feature2</i>	<i>Feature1</i>	<i>Feature2</i>
NN	6.5	5	16	15.5
IHGR	4	3.5	7.0	6.5
TFDA	3.5	3	4.5	4.5
JCBB	2	1.5	3.5	3

Table 5.8: The comparison of the computational time of several data association methods.

	Average CPU time of feature density of 0.008	Average CPU time of feature density of 0.0125
NN	0.0093	0.0148
IHGR	0.0165	0.0575
TFDA	0.0576	0.1855
JCBB	0.1069	0.2802

high.

5.4.2 Real Outdoor Environment

SLAM with Artificial Beacons

In order to implement the IHGR data association algorithm in a SLAM process in a real environment, we firstly use the experimental data set from [140] obtained by Guivant and Nebot. The testing site is a car park at Sydney University. The vehicle is equipped with a GPS, a laser sensor and wheel encoders. A kinematic GPS system of 2 cm accuracy was used to evaluate the ground truth. Thus, the true navigation map was available for comparison purposes. Wheel encoders give an odometric measurement of the vehicle location. The dead reckoning sensors and laser range sensor are combined together to predict the vehicle's trajectory using the extended Kalman filter and to build up the map at the same time. In this experiment, the features used are artificial beacons. The feature detection was done by using a geometric analysis of the range measurement to obtain the most likely centers of tree trunks using intensity information. The laser scans are processed using Guivant's algorithm [129] to detect tree trunks' centers and estimate their radii.

We ran continuous SLAM for more than 5000 time steps and obtained the map shown in Figure 5.5 using our proposed IHGR data association during the SLAM process. It can be seen that the IHGR method performs well for the SLAM implementation in real time. Figures 5.6 and 5.7 give the measurement (range and angle) innovation and their 2σ confidence bounds during the SLAM process, respectively. The results show that the IHGR based algorithm works well during the SLAM process.

In order to check the effectiveness of the IHGR data association, we randomly

Figure 5.5: The SLAM path and the feature map during the SLAM process with IHGR data association.

Figure 5.6: The range innovation and its 2σ confidence bounds during the SLAM process with IHGR data association.

Figure 5.7: The observation angle innovation and its 2σ confidence bounds during the SLAM process with IHGR data association.

choose two scans (scan 68 and scan 87) and show the matching matrix x_{nm} after the IP problem is solved. In scan 68, the laser sensor obtained 2 measurements and the existing feature number has accumulated to 11. As mentioned, the term “measurement” means the extracted features. As seen in Table 5.9, measurement 1 is associated with feature 3 and measurement 2 is associated with feature 2. The rest of the features are all undetected in this scan. In Table 5.10, for scan 87, measurement 1 is matched with a dummy element which means this is a new feature or false alarm. Since the probability of false alarm for the laser sensor is low, we regard the measurement as a new feature. The other measurements and features are similar to those in scan 68 which can be seen in Table 5.9. In this environment, the features are sparsely distributed. By checking the experimental data, we know that the number of observations per step ranges from 1 to 5. Therefore, the environment is relatively simple as far as data association is concerned. In this case, the NN algorithm also works well. We ran the whole SLAM with the NN algorithm and IHGR algorithm respectively and found that the computational cost

Table 5.9: The matching matrix in scan 68.

	Existed feature number and dummy element												
		1	2	3	4	5	6	7	8	9	10	11	0
scan 68	1	0	0	1	0	0	0	0	0	0	0	0	0
	2	0	1	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	1	1	1	1	1	1	1	1	0

Table 5.10: The matching matrix in scan 87.

		Existed feature number and dummy element														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	0
scan 87	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0

is similar although the NN algorithm is slightly faster than our method. For the NN data association, the whole SLAM process (involving more than 5000 time steps) where the vehicle travelled several hundred meters took 40.9680 seconds while the IHGR algorithm took 58.1420 seconds on the same computer(the algorithms were run on Pentium IV PC, 1.7GHz, and we calculated the CPU time for each algorithm).

SLAM with Natural Features in an Experimental Environment

In order to verify the effectiveness of the IHGR algorithm, a more complex real environment is explored by our car-like Cycab vehicle (see Figure 3.10). In this case, the SLAM is processed with natural features rather than artificial beacons in the last experiment. The features extracted from the complex campus environment are obtained by using the feature extraction algorithm given in the last chapter. The experimental data used was obtained from a 2-D scanning range laser (SICK LMS200) mounted in the front of the vehicle. The laser returns a 180 degree

planar sweep of range measurements in 0.5 degree intervals (i.e., 361 range values in an anti-clockwise order) with a range resolution of 50mm. Similar to the last experiment, the vehicle is equipped with a GPS, a laser sensor and encoders. A DGPS system is used to evaluate the ground truth. Thus, the true navigation map is available for comparison purposes. The testing site is a long walk way around Hall 7 at Nanyang Technological University of Singapore. The real campus map including this road is shown in Figure 5.8. The dead reckoning sensors and laser range sensor are combined together to predict the vehicle's trajectory using the extended Kalman filter and to build up the map at the same time. During SLAM, in order to improve our map accuracy, we attempted to detect two types of features, namely the point features and circular features as described in the last chapter and used the IHGR algorithm in Section 5.3 to carry out data association.

Figure 5.9 shows part of the experimental environment. Figure 5.10 gives a typical laser scan during the run and shows the main features that are extracted from the scan data. Figure 5.11 shows the experimental results of SLAM. In this figure, there is a break in the GPS data. This is because the vehicle is under a building which blocks the GPS signal (the GPS data coordinate is vacant roughly between $(-10, -120)$ and $(-30, -115)$). That is, no ground truth is available in this segment of road. From this figure, we can see that the estimated path is close to the GPS reading except the part of the road that is near to the break because the GPS reading is not accurate for this segment of the trajectory. Figures 5.12 and 5.13 indicate the observation innovations and their 2σ confidence bounds during the whole process which lasted for 450 seconds. These figures show that the SLAM process is consistent using the natural features. It is also important to mention that the average position error of the entire SLAM process except the segment where the ground truth is unavailable is smaller than 0.5 meter. On the other hand, we also carried out the SLAM process with the NN data association. Unfortunately, the NN algorithm results in a diverged vehicle path estimation as shown in Figure 5.14, due to the complex environment with features of high density. In Figure 5.14,

point A is the starting point of the SLAM process as in the previous experiment with the IHGR algorithm.

Figure 5.8: The map of our experiment site. The blue curve represents the vehicle’s trajectory; The starting and ending of the trajectory are indicated by red dots.

5.4.3 Discussions on the Proposed Algorithm

The proposed IHGR algorithm in this chapter is a global NN algorithm in essence. It considers one frame scan data and finds all possible hypothesis in the validation gate for the measurements and existing features. Compared to the two frame data association method, it has two advantages. It avoids solving an NP hard problem as is the case of the two frame data association method in [2, 3]. Certainly, the two frame data association can give more accurate and reliable matching results. The second advantage is that it only needs one step prediction and hence the effect of vehicle model uncertainty is smaller.

Figure 5.9: The environment and the vehicle in our experiment.

Figure 5.10: A typical laser scan of the experimental environment.

Figure 5.11: A comparison between the estimated path and the path by the GPS readings during SLAM.

Figure 5.12: The range observation's innovation and its 95% confidence bounds in the SLAM experiment.

Figure 5.13: The angular observation’s innovation and its 95% confidence bounds in the SLAM experiment.

Figure 5.14: The unsuccessful path estimation of this experiment when NN algorithm is applied. The vehicle started from point A in our experiment, see Figure 5.8.

5.5 Conclusions

This chapter presented a data association algorithm for SLAM which offers a good trade-off between accuracy and computational requirement. We first formulated the data association problem in SLAM as a two dimensional assignment problem. In our work, only 1 frame scan data is considered. The data association problem in SLAM is formulated as a two dimensional assignment problem rather than a 3 dimension one and is computationally more efficient. Further, since only one step prediction is involved, the effect of the vehicle model uncertainty is smaller as compared to the data association methods using two frame scan data. In order to obtain a fast solution, the 0-1 IP problem was firstly relaxed to an LP problem. Then we proposed to use the IHGR procedure in conjunction with basic LP algorithms to obtain a feasible solution of the data association problem. Both the simulation and experiment results demonstrated that the proposed algorithm is implementable and gives a better performance (higher successful rate) than the commonly used NN algorithm for complex (outdoor) environments with high density of features. Compared to the optimal JCBB algorithm, the proposed algorithm has lower computational complexity and is more suitable for real-time implementation.

Chapter 6

Entropy-based Method of Feature Selection

6.1 Introduction

A key stumbling block in the development and implementation of new methods for SLAM has been the computational requirement problem—the increase of computational complexity with the size of the operating environment of the mobile robot [103, 63].

In this section, the most closely related work to the proposed method is firstly reviewed. The idea in [112, 113, 114] is to reduce the computational complexity by a modified version of EKF. A compressed algorithm is presented in the references to store and maintain all the information gathered in a local area with a cost proportional to the square of the number of landmarks in the area. This information can be transferred to the rest of the global map with a cost of at most similar to full SLAM [112]. It mainly considers the system update problem from the aspect of filter technique. It is suitable for all estimation problems that can be processed by EKF besides SLAM.

The work in [103] presents a map management strategy for obtaining an efficient SLAM solution. It verifies that when a landmark is no longer visible from the robot's current location, it can be deleted from the map without affecting the consistency of the estimation process and it is indeed possible to remove a large percentage of the landmarks from the map without making the map building process inconsistent. The paper also shows that the landmark location estimates that are less correlated provide more information. There is no advantage to maintain the highly correlated landmarks in the map.

In this chapter, we present an entropy based feature selection strategy to reduce the computational cost for SLAM. The work in this chapter is closely related to the work [103]. Different from [103], we give a model based feature selection method. In essence, our method is to determine a selection that aims to maximize the total knowledge about the system in the presence of measurement and navigation uncertainty. The algorithm gives a compromise between computational requirement and quality estimations of vehicle pose and features' positions. The selection process is formulated as an optimization problem that maximizes the amount of information acquired. The information gained by observing an environmental feature must counteract the rise in uncertainty that results from the motion of the vehicle.

The chapter is organized as follows: Section 6.2 is devoted to the formulation of an entropy-based feature selection problem for SLAM. Section 6.3 presents a suboptimal approach to solve the formulated problem and integrate it with the map management in SLAM so as to obtain a performable SLAM implementation. Section 6.4 presents the map management for the efficient SLAM approach. Section 6.5 shows the simulation and experimental results. Some conclusions are drawn in Section 6.6.

6.2 Entropy-based Feature Selection for SLAM

6.2.1 Stochastic Mapping

We firstly review the theory of stochastic mapping from the point of view of standard Bayesian estimation theory. It is simply a special way of organizing the states in an EKF for the purpose of feature based navigation. The augmented state equation containing both the state of the vehicle and the states of all landmarks is rewritten as follows:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k) \quad (6.1)$$

where

$$\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_v(k), \mathbf{u}(k)) \\ p_1(k) \\ \vdots \\ p_N(k) \end{bmatrix}$$

and

$$\mathbf{v}(k) = [\mathbf{v}_v^T(k) \ 0 \ \cdots \ 0]^T.$$

The observation model for the i th landmark is written in the form

$$\mathbf{z}_i(k) = \mathbf{H}_i(\mathbf{x}(k), p_i(k)) + \mathbf{w}_i(k) \quad (6.2)$$

where the meaning of each variable has been defined in Chapter 3.

For convenience, we use $\mathbf{x}_{k+1}$ to represent $\mathbf{x}(k+1)$ in Equation (6.1).

The a posteriori PDF of $\mathbf{x}_{k+1}$, given a set of measurements $\mathbf{Z}_k$ which include measurements associated with all features, can be found by the Bayes Rule as:

$$p(\mathbf{x}_{k+1} | \mathbf{Z}_{k+1}) = \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})p(\mathbf{x}_{k+1} | \mathbf{Z}_k)}{p(\mathbf{z}_{k+1} | \mathbf{Z}_k)}. \quad (6.3)$$

The distribution $p(\mathbf{z}_{k+1} \mid \mathbf{x}_{k+1})$ is defined as the likelihood function by the Likelihood Principle. By knowing $p(\mathbf{x}_{k+1} \mid \mathbf{Z}_{k+1})$, we can form an estimate $\hat{\mathbf{x}}_{k+1}$ of the state. An EKF is an approximation of Bayesian formula. It is a computational efficient estimator for the states of a given nonlinear dynamic system and it assumes that the noise processes are well modelled by the Gaussian noise and that the error due to linearization of the nonlinear system is small. That is, the EKF for a system provides an estimate of both the state of the system $\hat{\mathbf{x}}$ as well as an estimate of the covariance $\mathbf{P}$ of the estimated state. The covariance of the estimated state provides an estimate of the confidence in the estimate $\hat{\mathbf{x}}$. The estimation error covariance $\mathbf{P}$ of the system takes the form of:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{rr} & \mathbf{P}_{r1} & \cdots & \mathbf{P}_{rn} \\ \mathbf{P}_{1r} & \mathbf{P}_{11} & \cdots & \mathbf{P}_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{nr} & \mathbf{P}_{n1} & \cdots & \mathbf{P}_{nn} \end{bmatrix}_{k|k}. \quad (6.4)$$

The sub-matrices $\mathbf{P}_{rr}$, $\mathbf{P}_{ri}$ and $\mathbf{P}_{ii}$, $i = 1, 2, \dots, n$, are the vehicle to vehicle, vehicle to feature, and feature to feature covariances, respectively. This form is significant as it allows us to separate the uncertainty associated with the robot as well as each individual feature. This separation will be used in obtaining a measure of the information in our system. Thus, the robot and the map are represented by a single state vector, $\mathbf{x}$, with an associated estimation error covariance $\mathbf{P}$ at each time step. As new features are added, $\mathbf{x}$ and $\mathbf{P}$ increase in size.

6.2.2 Problem Formulation

In the subsection, we model the entropy-based feature selection beginning with the standard Bayesian estimation theory. For simplicity, we rewrite (6.3) in another form:

$$p(\mathbf{x} \mid \mathbf{z}) = \frac{p(\mathbf{z} \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{z})}. \quad (6.5)$$

The system state $\mathbf{x}$ is the parameter to be estimated. The probability that the robot and the features have a pose $\mathbf{x}$ is given by $p(\mathbf{x})$. In Equation (6.5), $p(\mathbf{x} | \mathbf{z})$ is computed from sensor model and the robot's map of the environment. It may depend on two factors. The first is the noise corruption to the signal observed by the sensor. The second is the estimated system state $\mathbf{x}$. It includes the robot's location and features' locations estimated at each time step. These two factors bring the uncertainty to the sensor model.

Let $\mathbf{Z}$ be all measurements that have already been used in the inference of the current belief $p(\mathbf{x} | \mathbf{Z})$ of the robot and features' locations. Let z_i be an additional feature that hasn't been used so far, that is, z_i does not belong to $\mathbf{Z}$. The fusion of the additional feature z_i and the current system state belief $p(\mathbf{x} | \mathbf{Z})$ yields the next system state belief $p(\mathbf{x} | z_i, \mathbf{Z})$ as:

$$p(\mathbf{x} | z_i, \mathbf{Z}) = Cp(z_i | \mathbf{x}, \mathbf{Z})p(\mathbf{x} | \mathbf{Z}) \quad (6.6)$$

where C is a normalization constant. We assume that $\mathbf{Z}$ and z_i are independent conditioned on $\mathbf{x}$, that is, $p(z_i | \mathbf{x}, \mathbf{Z}) = p(z_i | \mathbf{x})$, then Equation (6.6) becomes:

$$p(\mathbf{x} | z_i, \mathbf{Z}) = Cp(z_i | \mathbf{x})p(\mathbf{x} | \mathbf{Z}). \quad (6.7)$$

According to the information theory [141], the uncertainty of an arbitrary continuous distribution $p(x)$ can be described by its entropy:

$$H(p_x) = - \int p(\mathbf{x}) \log(p(\mathbf{x})) d\mathbf{x}. \quad (6.8)$$

This measure can be considered as the “purity” of the probability distribution. If the distribution is highly focused at a single pose $\mathbf{x}$, then the entropy will be low. If the distribution is spread over a wide space, the entropy will be high. According

to Equation (6.8), the effect that a particular set of sensing data has on the robot's and features' belief in their positions can be measured in this way.

$$H(p_{x|Z}) = - \int p(\mathbf{x} | Z) \log(p(\mathbf{x} | Z)) d\mathbf{x}. \quad (6.9)$$

Combining Equation (6.5) and Equation (6.9), they give the entropy of the posterior distribution after obtaining the measurements:

$$H(p_{x|Z}) = - \int \frac{p(\mathbf{z} | \mathbf{x})p(\mathbf{x})}{p(\mathbf{z})} \log\left(\frac{p(\mathbf{z} | \mathbf{x})p(\mathbf{x})}{p(\mathbf{z})}\right) d\mathbf{x}. \quad (6.10)$$

This equation gives the entropy of the posterior distribution before additional features are added. Recall that $p(\mathbf{x})$ is the prior position distribution and $p(\mathbf{z} | \mathbf{x})$ is the probability of the sensor measurement conditioned on the position, computed from the sensor model and the environmental map.

If we use an additional feature to update the system state, the uncertainty of the next system state $p(\mathbf{x} | z_i, \mathbf{Z})$ is a function of both $\mathbf{Z}$ and the additional feature measurement z_i , and can be described by the entropy $H(p_{x|z_i, \mathbf{Z}})$ as:

$$H(p_{x|z_i, \mathbf{Z}}) = - \int p(\mathbf{x} | z_i, \mathbf{Z}) \log(p(\mathbf{x} | z_i, \mathbf{Z})) d\mathbf{x}. \quad (6.11)$$

We want to find the additional features that can give more information gain if they are indeed incorporated in the fusion with the existing features' and robot's position distribution.

$$i = \arg \max_{i \in I} (H(p_{x|Z}) - H(p_{x|z_i, Z})).$$

6.3 Entropy Based Feature Selection Strategy

The essence of our model in the last section is to determine which feature should be selected to update the system state in the SLAM so as to maximize the total knowledge (i.e., the information) about the system in the presence of measurement and navigational uncertainty. By choosing features, we mean that some “good” features are chosen so as to maximize the information gain about the robot’s location and the locations of all the features (the map).

The proposed approach for the selection of a particular landmark is based on localization and mapping information offered by a particular landmark from a given time step. It takes into account the uncertainty in the vehicle pose estimate and the existing features’ positions in the map while computing the information content of the landmark.

The concept of the entropy is employed to facilitate landmark augmentation. As stated in the last section, the entropy $H(\cdot)$ is a measurement of the average uncertainty of a random variable and thus represents the compactness of the probability distribution. Therefore, it is a measure of the informativeness of the distribution where information is defined as the negative of entropy. The entropy is minimum when the information is maximum. It is conventional to seek minimal entropy when actually maximum information is sought. A mathematical expression for the entropy of a Gaussian distribution is to be used in the following. For an n dimensional state vector $\mathbf{x}_{k+1}$ conditioned on an observation vector $\mathbf{Z}_{k+1}$, the posterior entropy can be obtained as:

$$H_{k+1|k+1} = E\{-\ln p(\mathbf{x}_{k+1} | \mathbf{Z}_{k+1})\}. \quad (6.12)$$

For a system state of a Gaussian distribution, we can calculate the value of the above equation [141]

$$H_{k+1|k+1} = 0.5 \ln[(2\pi e)^n | \mathbf{P}_{k+1|k+1} |]. \quad (6.13)$$

Therefore, for a Gaussian distribution vector all that is required to compute its entropy is the length n and covariance $\mathbf{P}$.

The posterior and prior information metrics can then be defined as

$$im_{k+1|k+1} = -H_{k+1|k+1} = -0.5 \ln[(2\pi e)^n | \mathbf{P}_{k+1|k+1} |], \quad (6.14)$$

$$im_{k+1|k} = -0.5 \ln[(2\pi e)^n | \mathbf{P}_{k+1|k} |]. \quad (6.15)$$

Thus, we can calculate the information difference as follows:

$$\Delta I = im_{k+1|k+1} - im_{k+1|k}. \quad (6.16)$$

In our application, when some observations are coming at some time step, we will perform the following actions to select the useful features which can provide us more information for the vehicle's pose and map features' positions. We firstly calculate the individual update covariance $\mathbf{P}_{k+1,i|k+1,i}$. This means we use only one feature at time step $k+1$ to update the system to get a covariance $\mathbf{P}_{k+1,i|k+1,i}$ by using Equations (3.10) and (3.14). In view of (6.14)–(6.16), we can calculate the individual information gain of using each feature to update the estimation as follows:

$$\Delta I_i = im_{k+1|k+1,i} - im_{k+1|k}, \quad i = 1, 2, \dots, N \quad (6.17)$$

where

$$im_{k+1|k+1,i} = -0.5 \ln[(2\pi e)^n | \mathbf{P}_{k+1,i|k+1,i} |] \quad (6.18)$$

In the above formula, we only show how to calculate the information gain for one feature. We note that there exist correlations among features which should

be considered in order to maximize the information gain when selecting features. Based on the available resources, we assume a fixed number of features are to be chosen at each time step. Therefore, a combination of features is considered when we calculate the information difference for different combinations of the features. For example, if we want to select N_0 features from N features, there will be $C_N^{N_0} = \frac{N!}{N_0!(N-N_0)!}$ combinations, where $N_0! = 1 \times 2 \times \dots \times N_0$. For each combination, we can calculate the value of the information gain of the set of features similar to the method for one feature that is denoted by the above formula.

However, the computational complexity of this approach prevents its practical applicability. In order to overcome this problem, a heuristic suboptimal approach is proposed.

From [103], it is known that the feature location estimates that are less correlated provide more information. In the limiting case, when two landmark location estimates are fully correlated, there is no advantage to be gained by maintaining both these features in the map except geometric considerations. Thus, when selecting the features of the combination which provides the maximum information gain, we should select those features that are less correlated with those features that have already been selected and have large individual gain. This is performed as follows:

- Step 1: Initialize the set of selected features by first choosing a feature from the candidate features that gives rise to the maximum individual information gain;
- Step 2: From the remaining candidate features that have not been selected, select the one that has the maximum individual information gain. If the selected one has weak correlations with those features that have already been selected, we add it to the set of selected features. Otherwise, we ignore it.
- Step 3: Repeat Step 2 until the number of the selected features reach N_0 .

6.4 Map Management

Now we focus on the map management problem. As we know, once features have been detected, they must be matched against known landmarks in the environment. Data association is firstly performed between the observed features and the features currently in the map. This step is one of the most crucial step in the mapping process. When data is received from a sensor there is a possibility that it may in fact be a spurious measurement. Some spurious measurements can be eliminated by the development of appropriate feature extraction approach or robust feature extractor [103]. Regardless of the care that is taken in designing the feature extractor, some spurious measurements may still be passed to the localization and mapping algorithm and it is important to have a mechanism for rejecting these. Here we adopt the strategy as in [83]. A new feature is initialized when an observation can not be matched with one of the existing features. The “old” features that are not observed for a number of successive scans will be deleted. But the NN data association is replaced by an more effective matching algorithm of the last chapter in order to reduce the wrong matchings. In detail, the data association and map management can be summarized as follows:

- Matching: The features and the observations are matched by the IHGR method proposed in the last chapter, and measurements that have not been matched are stored.
- Feature selection: Select features according to the proposed heuristic approach. The number of selected features can be decided by a proper compromise between quality estimation of vehicle pose and available computation resource.
- Updating system state: Update the vehicle and features’ positions with the selected features by EKF.
- Initializing new feature: New features which are not matched with the features stored in the map by the IHGR algorithm are initialized.

- Removing old features: Out-of-date features are deleted.

6.5 Simulation and Experimental Results

6.5.1 Simulation Results

The first test environment is established by randomly generating some features and assuming that the vehicle's trajectory is a circle whose radius is 62 meters. The robot moves at a constant speed and the heading angle changes 1 degree at each sampling instant. The environment involves 160 features which are randomly distributed in a region of 150 meters by 150 meters. When the vehicle moves, some of the features are observed. We assume that the features' positions are unknown which is the case in SLAM and match the features with the observations by using the IHGR algorithm in Chapter 5.

By applying the entropy based feature selection scheme proposed in Section 6.3, we implement the SLAM and the results are shown in Figure 6.1. Here, the number of features to be selected is 10 (we assume that there are 18 observations in each scan) after a fixed time interval (20 laser scans). Figure 6.1 shows the SLAM results that include the vehicle's estimated trajectory and the features' positions. Circles stand for the features that appeared during the full SLAM and the stars stand for the features selected during the SLAM process by using our entropy based feature selection scheme. In fact, the vehicle's true path is almost overlapped with the estimated one during the SLAM process. In Table 6.1, we show the entropy calculation results for the features in time step 437. The first column means the sequence number according to the information difference value (see the second column in Table 6.1) for each observation of the scan. The value of the third column in Table 6.1 is the corresponding feature numbers stored in the system state. We select the top 10 features which have the information difference Δi

Figure 6.1: The mapping and vehicle path when using the proposed entropy based feature selection in the SLAM process.

Figure 6.2: The vehicle’s position errors (between the ground truth (GPS reading) and the estimated position) comparison in the global coordinate between the entropy based feature selection in the SLAM process and full SLAM.

Figure 6.3: The range innovation and its 2σ confidence bounds during the SLAM process with the proposed method.

Figure 6.4: The angle innovation and its 2σ confidence bounds during the SLAM process with the proposed method.

Table 6.1: The information difference for each observation in Scan 437.

Sequence number	Information difference ΔI	Feature number	The maximum correlations with the selected features $\bar{p}$
1	4.8764	18	–
2	4.6243	12	0.0007
3	4.3446	17	0.0004
4	4.2218	9	0.0006
5	3.9777	131	0.0071
6	3.7322	3	0.0081
7	3.5023	16	0.0023
8	3.3436	1	0.0015
9	3.2120	14	0.0075
10	3.0232	15	0.0027
11	2.8551	8	0.0174
12	2.7067	10	0.0206
13	2.5532	5	0.0564
14	2.3735	11	0.0675
15	2.8338	2	0.0812
16	1.9611	6	0.0315
17	1.6566	4	0.0254
18	1.2621	7	0.0645

greater than 3.0000. The fourth column of the table shows the rough estimation of the maximum correlation between the features that have been selected and the candidate feature to be considered. The relative value of the correlation $\bar{p}$ is calculated as follows: $\bar{p} = \frac{Max\{trace(P_{ij})\}}{trace(P_{ii})}$ where $trace(P_{ij})$ means the trace of matrix P_{ij} which is the cross covariance matrix between the position estimate of the candidate feature i and the selected feature j . P_{ii} is the variance matrix of the position estimate of feature i . From this table, we can see the the selected features have very weak correlations.

The global error between the estimated vehicle position and the ground truth can be seen in Figure 6.2 where the comparison between the global error with our proposed feature selection algorithm and the global error of using all the obtained

Table 6.2: SLAM performance and the algorithm computational complexity versus the number of the selected features.

The num- ber of selected features	Errors(Meters)		time
	X direction er- ror	Y direction er- ror	
2	0.2285	0.2508	65.8914
5	0.1809	0.1908	69.3498
8	0.1547	0.1789	75.5784
10	0.1319	0.1619	80.7560
13	0.1275	0.1528	83.1237
15	0.1021	0.1222	86.4489
18	0.0771	0.0898	91.5482

features is made. From the figure, we can find that the feature selection algorithm gives good results as compared to the full feature (using all features) SLAM and with a lower computational cost. For the full SLAM, the whole SLAM process where the vehicle travelled about two hundred meters took 91.5482 seconds while the proposed algorithm took 80.7560 seconds with 10 features chosen on the same computer(the algorithms were run on Pentium IV PC, 1.7GHz).

In the feature selection algorithm, suitable choice of the number of selected features and the time interval for updating the selection depend on the range, accuracy and the speed of the sensor used, process noise and the trade-off between the localization accuracy required and the computational efficiency. Thus, here we give some simulations to show the comparison of the computational complexity of the whole SLAM process (here we use execution time of the algorithm in the same computer to represent it) when the number of the selected features is changed. And we also show the comparison of the vehicle pose estimation error when the number of selected features is changed every fixed time interval.

In Table 6.2, the errors (X direction error and Y direction error) are the same defined as those in Equations (5.11) and (5.12).

6.5.2 Experimental Results

In this subsection, a practical implementation of the proposed entropy-based feature selection scheme in an outdoor environment in Nanyang Technological University is presented. The testing site is a road around Canteen 3 and Hall 7 at Nanyang Technological University. Many trees and pillars exist on both sides of the road. Figure 6.5 shows the beginning part of the experimental environment. The vehicle is equipped with a GPS, a laser sensor and wheel encoders. A kinematic GPS system of 2 cm accuracy is used to evaluate the ground truth. Thus, the true navigation map is available for comparison purposes. Wheel encoders give an odometric measurement of the vehicle location. The dead reckoning sensors and laser range sensor are combined together to predict the vehicle's trajectory using the EKF and to build the map at the same time. In this experiment, the features used are natural landmarks (mainly the trees, pillars on the road and edges) that are extracted by applying the feature detection algorithm in Chapter 4. The feature detection was done by using the curve gradient model for data segmentation and the Gauss-Newton optimization method for parameter acquisition.

Figure 6.6 shows the real map of the experimental environment where the mobile robot travelled. Figure 6.7 gives the comparison of the estimated trajectory of the vehicle and the true path. It also gives the features extracted from the raw laser sensor data which is represented by the green circles and the selected features which are marked by red stars and applied for the SLAM update.

Absolute error estimates within 0.8 meter are achieved in the position estimates. Figures 6.8 and 6.9 give the validated range and bearing innovation sequences along with the 3σ bounds. It is evident that the innovations of range and bearing are well within the bounds, which shows that the estimate is consistent.

Figure 6.5: Part of the experimental site (we started from this point).

Figure 6.6: The real map of the vehicle trajectory at Nanyang Technological University.

Figure 6.7: The experimental results. The figure shows the vehicle’s estimation path (red line), the features extracted (green circles) and the selected features (red star) by using the entropy based feature selection algorithm during the SLAM process.

Figure 6.8: The experimental results. The range innovation and its 3σ confidence bounds during the SLAM process with the entropy based SLAM.

Figure 6.9: The experimental results. The observation angle innovation and its 3σ confidence bounds during the SLAM process with the entropy based SLAM.

6.6 Conclusions

This chapter considered the problem of selecting features for feature-based SLAM for autonomous outdoor vehicles. A suboptimal feature selection strategy was incorporated within a stochastic mapping algorithm and tested via simulations and real data in an outdoor environment. We introduced a method for performing SLAM in a priori unknown environments with large number of features. The method is based on choosing features that, given the current knowledge, would maximize the information gain in the estimation of the vehicle's pose and the features' position. We can select the number of features to update the SLAM state according to the availability of computational facility. This approach can easily be implemented as an extra step in a stochastic mapping algorithm for SLAM. The validity and usefulness of the approach has been verified by simulations and experiments. However, the relative geometry of the features is not considered in the algorithm, this should be an important aspect for feature selection in the

futures' research.

Chapter 7

Conclusions and Recommendations for Further Research

7.1 Conclusions

In this thesis, we have investigated the algorithms development for feature extraction, data association and computational cost reduction of SLAM systems. To summarize the main results of this dissertation, we point out the following aspects.

We have addressed the problem of extracting edge and circle features for semi-structured outdoor mobile robot navigation. We classified features into edges, circles and lines and proposed two approaches to the extraction of edge and circle features. Experimental results showed that the proposed two approaches are efficient in detecting features for SLAM in semi-structured outdoor environments.

For more general outdoor environments, we have introduced a curve gradient model for smooth surfaces for data segmentation and determination of feature type. There

is one main advantage of the proposed method—the adaptive characteristic for any shape curve data recorded from an arbitrary smooth surface, such as building walls, trees, complex irregular rock or bush surfaces. The effectiveness of the feature detection method has been verified by simulations and experiments.

We have presented an efficient IP based data association approach for SLAM. In this approach, the feature based SLAM data association problem is formulated as a 0-1 IP problem. This makes it possible to solve the complex problem by different heuristic optimization algorithms which have been developed in recent years. The proposed algorithm deals with the global matching between existing features and measurements of each scan and is more robust for an environment of high density features (the feature number is high and the distances between features are often very close) which is usually the case in outdoor applications. Experimental results show that the proposed algorithm has more superiority for data association than the NN algorithm in the environments where the feature density is high. Compared to the optimal algorithms, such as the JCBB and TFDA, it has lower computational requirement that makes it more suitable for real time implementation.

In order to reduce the computational burden during real time SLAM for outdoor mobile robot navigation, we have proposed an information based method of feature selection. Our method has the advantage of selecting the number of features by considering the computational constraint in real time implementation. This algorithm is capable of maintaining a manageable computational cost which is decided during the SLAM implementation.

We also designed and implemented the real time autonomous vehicle localization system by combining the software and hardware together with the communication system of the mobile robot. All the proposed algorithms have been verified by real outdoor environments with the mobile robot system.

7.2 Recommendations for Further Research

There are still some problems to be further investigated. In this section, we recommend the following promising topics for further research.

- It remains interesting to investigate feature extraction methods for completely unstructured outdoor environments. For these environments, point features do not give sufficient information to guarantee a proper localization. Techniques that describe the terrain surface are investigated nowadays [77]. Probably the combination of both different environment description concepts will give the best results in efficiency and robustness. Another challenging topic for feature detection is the 3-D laser scanner feature extraction of SLAM.
- Dynamic environments are also a main problem for SLAM. SLAM processes should be able to manage changing environments. Environments that are modified are typical in nature and the human world. Especially, in outdoor environments, there are often some uncertainties, such as moving vehicles, people, and wind. Therefore, how to develop algorithms to adapt the dynamic environments is a big challenge.
- It is very interesting if we can use all of the available information sensed from the sensors. An efficient navigation system must incorporate any useful source of information. At present sensors such as video, GPS, compass, etc. can be used in the SLAM process in addition to the traditional dead-reckoning and scanning sensors.
- In unstructured outdoor environments, another difficult problem is process modelling. Outdoor SLAM needs to contend with 2-D and 3-D environments, and rugged terrains. Thus, it is necessary to develop process models that represent high-speed, non-planar motion and high levels of slip.

Author's Publications

- [1] S. Zhang, L. Xie, and M. D. Adams, "An efficient data association approach to simultaneous localization and map building," *International Journal of Robotics Research*, Vol. 24, No. 1, pp. 49–60, 2005
- [2] S. Zhang, L. Xie, and M. D. Adams, "Feature Extraction for Outdoor Mobile Robot Navigation Based on a Modified Gauss-Newton Optimization Approach," *Robotics and Autonomous Systems (journal)*, Vol. 54, Issue 4, pp. 277–287, 2006
- [3] S. Zhang, L. Xie, and M. D. Adams, "Outdoor mobile robot localization based on a novel feature extraction method and particle filter," *the second revised version submitted to journal for possible publication*.
- [4] S. Zhang, L. Xie, and M. D. Adams, "An efficient data association approach to simultaneous localization and map building," *Proc. of IEEE Int. Conf. Robot. Automat.*, New Orleans, USA, April 2004, pp. 1493–1498.
- [5] M. D. Adams, S. Zhang, and L. Xie, "Particle filter based outdoor robot localization using natural features extracted from laser scanner," *Proc. of IEEE Int. Conf. Robot. Automat.*, New Orleans, USA, April 2004, pp. 854–859.
- [6] S. Zhang, L. Xie, and M. D. Adams, "Gradient model based feature extraction for simultaneous localization and mapping in outdoor applications," *Proc. of the Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV 2004)*, Kunming, China, Dec. 2004, pp. 431–436.

-
- [7] S. Zhang, L. Xie, and M. D. Adams, "Geometrical feature extraction using 2D range scanner," *Proc. of the Fourth Int. Conf. Contr. and Automat.*, Montreal, Canada, June 2003, pp. 901–905.
- [8] S. Zhang, L. Xie, and M. D. Adams, "Entropy-based feature selection scheme for simultaneous localization and map building in outdoor environments," *Proc. of IEEE/RSJ Int. Conf. Intelligent Robot. Systems (IROS)*, Edmonton, Canada, August 2005, pp. 1175–1180.
- [9] S. Zhang, L. Xie, and M. D. Adams, "Feature extraction for outdoor mobile robot navigation with laser range sensor," *Nanyang Technological University Research Bulletin*, 2003, pp. 84-85.
- [10] S. Zhang, L. Xie, and M. D. Adams, "Entropy-based feature selection scheme for simultaneous localization and map building in outdoor environments," *to be submitted to journal*.

Bibliography

- [1] H. F. Durrant-Whyte, S. Majunder, M. Batista, and S. Scheduling, A bayesian algorithm for simultaneous localisation and map building, in *Proceedings of the International Symposium of Robotics Research*, Lorne Victoria, Australia, November 2001, pp. 3118–3123.
- [2] D. L. Perera, W. S. Wijesoma, and M. D. Adams, Multi-dimensional assignment based data association for simultaneous localization and mapping of autonomous vehicles, in *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS)*, Singapore, December 2003.
- [3] D. L. Perera, W. S. Wijesoma, and M. D. Adams, On multidimensional assignment data association for simultaneous robot localization and mapping, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, April 26-May 1 2004, pp. 860–865.
- [4] H. P. Moravec and A. Elfes, High resolution maps from wide angle sonar, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, Sept. 1985, pp. 116–121.
- [5] T. Bailey., *Mobile robot localization and mapping in extensive outdoor environment*, Ph.D thesis, Sydney Univ., 2002.
- [6] M. D. Adams, On-line gradient based surface discontinuity detection for outdoor scanning outdoor sensors, in *Proceedings of IEEE/RSJ International*

- Conference on Robotics and Systems*, Maui, Hawaii, October 2001, pp. 1726–1731.
- [7] J. J. Leonard and H. F. Durrant-Whyte, Underwater sonar data fusion using an efficient multiple hypothesis algorithm, in *Proceedings of IEEE International Conference on Robots and automation*, Minnesota, April 1995, pp. 2995–3002.
- [8] T. Tsumura, Survey of autonomous guided vehicles in japanese factory, in *Proceedings of IEEE International Conference on Robots and automation*, San Fransisco, CA, April 1986, pp. 1329–1334.
- [9] D. Polmerlau, Ralpth: rapidly adapting lateral position handler, in *Proceedings of IEEE International Symposium on Intelligent Vehicles*, Detroit, September 1995, pp. 506–511.
- [10] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, Mobile robot positioning sensors and techniques, *Journal of Robotic Systems*, vol. 14, pp. 231–249, 1997.
- [11] J. Borenstein and L. Feng, A new method for combining data from gyros and odometry in mobile robots, in *Proceedings of IEEE International Conference on Robots and automation*, Minnesota, April 1996, pp. 423–428.
- [12] H. F. Durrant-Whyte, An autonomous guided vehicle for cargo handling applications, *International Journal of Robotics Research*, vol. 45, pp. 407–440, 1996.
- [13] P. Weckesser and R. Dillman, Modelling unknown environment with a mobile robot, *Robotics and Autonomous Systems*, vol. 23, pp. 293–300, 1998.
- [14] E. M. Nebot, H. F. Durrant-Whyte, and S. Scheduling, Frequency domain modelling of aided gps for vehicle navigation systems, *Robotics and Autonomous Systems*, vol. 25, pp. 73–82, 1998.
- [15] H. Evers and G. Kasties, Differential gps in a real-time land vehicle environment - satellite based van carrier location system, *IEEE Aerospace and Electrical Systems Magazine*, vol. 9, pp. 26–32, 1994.

- [16] J. Leonard and H. F. Durrant-Whyte, Mobile robot localisation by tracking geometric beacons, *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 376–382, 1991.
- [17] W. Burgard, D. Fox, and D. Hennig, Fast grid-based position tracking for mobile robots, in *Proceedings of the 21th German Conference on Artificial Intelligence*, Germany, April 1997, Springer Verlag.
- [18] D. Fox, W. Burgard, and S. Thrun, Active markov localization for mobile robots, *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.
- [19] S. Panzieri, F. Pascucci, and G. Ulivi, An outdoor navigation system using gps and inertial platform, *IEEE/ASME Transactions on Mechatronics*, vol. 7, pp. 134–142, 2002.
- [20] D. Burschka and G. D. Hager, V-gps image-based control for 3d guidance systems, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Taiwan, Aug. 2003, pp. 1789–1795.
- [21] R. Sim and G. Dudek, Mobile robot localization from learned landmarks, in *IEEE/RSJ International Conference on Intelligent Robots and System*, Victoria, BC, Canada, Oct. 1998, pp. 1060–1065.
- [22] L. Zhang and B. K. Ghosh, Geometric feature based 2d map building and planning with laser, sonar and tactile sensors, in *Proceedings of the Fifth World Congress on Intelligent Control and Automation Artificial Intelligence*, Germany, Nov. 2000, pp. 115–120.
- [23] F. Lorussi, A. Marigo, and A. Bicchi, Optimal exploratory paths for a mobile rover, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, April 2001, pp. 2078–2083.
- [24] Yingjie Sun, Qixin Cao, and J. Lee, Robust mobile robot on-the-fly localization: using geometrical feature matching method, in *Proceedings of the Fifth World Congress on Intelligent Control and Automation Artificial Intelligence*, Germany, June 2004, pp. 4749–4753.

- [25] P. Jensfelt, D. J. Austin, O. Wijk, and M. Andersson, Feature based condensation for mobile robot localization, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, April 2000, pp. 2531–2537.
- [26] J. S. Esteves, A. Carvalho, and C. Couto, Generalized geometric triangulation algorithm for mobile robot absolute self-localization, in *Proceedings of IEEE International Symposium on Industrial Electronics*, Victoria, BC, Canada, June 2003, pp. 346–351.
- [27] D. Fox, W. Burgard, and S. Thrun, Markov localization for mobile robots in dynamic environments, *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 2000.
- [28] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, Robust monte carlo localization for mobile robots, *Artificial Intelligence*, vol. 128, pp. 1–2, 2001.
- [29] N. Bergman, L. Ljung, and F. Gustafsson, Terrain navigation using bayesian statistics, *IEEE Control Systems Magazine*, vol. 19, pp. 33–40, 1999.
- [30] J. Borenstein and L. Feng, Correction of systematic odometry errors in mobile robots, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Pittsburgh, PA, Aug. 1995, pp. 569–574.
- [31] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, The aiding of a low cost, strapdown inertial unit using modelling constraints in land vehicle applications, *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 731–747, 2001.
- [32] S. Cooper and H. F. Durrant-Whyte, A kalman filter model for gps navigation of land vehicles, in *Proceedings of IEEE International Conference on Robotics and Automation*, San Diego, CA, April 1994, pp. 157–163.
- [33] T. Aono, K. Fujii, S. Hatsumoto, and T. Kamiya, Positioning of vehicle on undulating ground using gps and dead reckoning, in *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, April 1998, pp. 3443–3448.

- [34] C. C. Lin and R. L. Tummala, Mobile robot navigation using artificial landmarks, *Journal of Robotics Systems*, vol. 14, pp. 93–106, 1997.
- [35] S. Thrun, Finding landmarks for mobile robot navigation, in *Proceedings of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, April 1998, pp. 958–963.
- [36] J. J. Leonard and H. F. Durrant-Whyte, Dynamic map building for an autonomous mobile robot, *International Journal of Robotics Research*, vol. 11, pp. 286–298, 1992.
- [37] K. Chong and L. Kleeman, Mobile robot map building from an advanced sonar array and accurate odometry, *International Journal of Robotics Research*, vol. 18, pp. 20–36, 1999.
- [38] G. Kantor and S. Singh, Preliminary results in range-only localization and mapping, in *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 1818–1823.
- [39] A. J. Davison and D. W. Murray, Simultaneous localization and map-building using active vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 865–880, 2002.
- [40] A.I. Eliazar and R. Parr, Dp-slam 2.0, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, USA, May 2004, pp. 1314–1320.
- [41] S. B. Williams, P. Newman, G. Dissanayake, and H. F. Durrant-Whyte, Autonomous underwater simultaneous localisation and map building, in *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 2000, pp. 1793–1798.
- [42] E. Jose and M. D. Adams, Millimetre wave radar spectra simulation and interpretation for outdoor slam, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, USA, May 2004, pp. 1321–1326.

- [43] A. Davidson, *Mobile robot navigation using active vision*, Ph.D Thesis, University of Oxford, 1999.
- [44] G. S. Lionis and K. J. Kyriakopoulos, A laser scanner based mobile robot slam algorithm with improved convergence properties, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Lausanne, Switzerland, Sept. 2002, pp. 582–587.
- [45] F. Dellaert and A. W. Stroupe, Linear 2d localization and mapping for single and multiple robot scenarios, in *Proceedings of IEEE International Conference on Robotics and Automation*, Washington DC, USA, May 2002, pp. 688–694.
- [46] C. Wang, C. Thorpe, and S. Thrun, Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas, in *Proceedings of IEEE International Conference on Robotics and Automation*, Taiwan, May 2003, pp. 842–849.
- [47] C. Wang and C. Thorpe, Simultaneous localization and mapping with detection and tracking of moving objects, in *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 2918–2924.
- [48] T. Baily, Constrained initialisation for bearing-only slam, in *Proceedings of IEEE International Conference on Robotics and Automation*, Taiwan, Sept. 2003, pp. 1966–1971.
- [49] G. Dissanayake, M. Hebert, A. Stentz, and H. F. Durrant-Whyte, Map building and terrain aided localisation in an underground mine, in *Proceedings of the International Conference on Field and Service Robotics*, Canberra, December 1997, pp. 34–40.
- [50] S. Scheding, G. Dissanayake, E. Nebot, and H. Durrant-Whyte, Autonomous navigation of an underground mining vehicle, *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 85–95, 1999.

- [51] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, Towards multi-vehicle simultaneous localisation and mapping, in *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002, pp. 2743–2748.
- [52] S. B. Williams, G. Dissanayake, and H. F. Durrant-Whyte, Terrain-aided navigation for underwater robotics, *RSJ Advanced Robotics*, vol. 15, pp. 533–550, 2001.
- [53] O. Wulf, K. Arras, H. Christensen, and B. Wagner, 2d mapping of cluttered indoor environments by means of 3d perception, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, USA, May 2004, pp. 4204–4209.
- [54] C. Brenneke, O. Wulf, and B. Wagner, Using 3d laser range data for slam in outdoor environments, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Las Vegas, USA, Oct. 2003, pp. 188–193.
- [55] R. Smith, M. Self, and P. Cheeseman, A stochastic map for uncertain spatial relationships, in *Proceedings of the Fourth International Symposium of Robotics Research*, California, USA, July 1987, pp. 467–474.
- [56] P. Moutarlier and R. Chatila, Stochastic multisensory data fusion for mobile robot location and environmental modelling, in *Proceedings of the Fifth International Symposium of Robotics Research*, Tokyo, June 1989, pp. 85–94.
- [57] C. M. Smith and J. J. Leonard, A multiple-hypothesis approach to concurrent mapping and localization for autonomous underwater vehicles, in *Proceedings of IEEE International Conference on Field and Service Robotics*, Minnesota, December 1997, pp. 2995–3002.
- [58] S. Betge-Brezet, P. Hebert, R. Chatila, and M. Devy, Uncertain map making in natural environments, in *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 1048–1053.

- [59] J.J. Leonard and H. F. Durrant-Whyte, Simultaneous map building and localization for an autonomous mobile robot, in *Proceedings of IEEE International Workshop on Intelligent Robotics and System*, Osaka, November 1991, pp. 1442–1447.
- [60] J. Neira and J. D. Tardos, Data association in stochastic mapping using the joint compatibility test, *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 890–897, 2001.
- [61] H. Choset and K. Nagatani, Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization, *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 125–137, 2001.
- [62] S. B. Williams, G. Dissanayake, and H. Durrant-Whyte, Constrained initialisation of the simultaneous localisation and mapping algorithm, in *Proceedings of International Conference on Field and Service Robotics*, Helsinki, Finland, June 2001, pp. 315–330.
- [63] S. B. Williams, G. Dissanayake, and H. F. Durrant-Whyte, Efficient solutions to autonomous localisation and mapping, in *Proceedings of the 15th IFAC World Congress*, 2002.
- [64] G. Giralt, R. Sobek, and R. Chatila, A multi-level planning and navigation system for a mobile robot: a first approach to hilare, in *Proceedings of International Joint Conference on Artificial Intelligence*, July 1979, pp. 335–337.
- [65] C. F. Olson, Subpixel localization and uncertainty estimation using occupancy grids, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, May 1999, pp. 1987–1992.
- [66] A. A. Makarenko, S. B. Williams, and H. F. Durrant-Whyte, Decentralized certainty grid maps, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, USA, Oct. 2003, pp. 3258–3263.

- [67] D. Wolf and G. S. Sukhatme, Online simultaneous localization and mapping in dynamic environments, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, April 2004, pp. 1301–1307.
- [68] P. Stepan, M. Kulich, and L. Preucil, Robust data fusion with occupancy grid, *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 35, pp. 106–115, 2005.
- [69] E. Ivanjko and I. Petrovic, Extended kalman filter based mobile robot pose tracking using occupancy grid maps, in *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference*, USA, May 2004, pp. 311–314.
- [70] C. Jennings, D. Murray, and J. J. Little, Cooperative robot localization with vision-based mapping, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, May 1999, pp. 2659–2665.
- [71] A. Eliazar and R. Parr, Dp-slam: fast, robust simultaneous localization and mapping without predetermined landmarks, in *The Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, August 2003.
- [72] A. Eliazar and R. Parr, Dp-slam 2.0, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, April 26–May 1 2004.
- [73] J. Nieto, J. Guivant, and E. Nebot, The hybrid metric maps (hymms): a novel map representation for denseslam, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, April 26–May 1 2004.
- [74] A. Stevens, M. Stevens, and H. F. Durrant-Whyte, Oxnav: reliable autonomous navigation, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, May 1995, pp. 2607–2612.
- [75] J. Guivant, E. M. Nebot, and S. Baiker, Localization and map buiding using laser range sensors in outdoor applications, *Journal of Robotic Systems*, vol. 17, pp. 565–583, 2000.

- [76] E. N. Macleod and M. Chiarella, Navigation and control breakthrough for automation technology, in *Proceedings of SPIE Conference on Mobile Robots*, Boston, April 1993, pp. 57–68.
- [77] S. Majumder, *Sensor fusion and feature based navigation for subsea robots*, Phd Thesis, The University of Sydney, 2001.
- [78] J. L. Crowley, Navigation for an intelligent mobile robot, *IEEE Transactions on Robotics and Automation*, vol. RA-1, pp. 31–41, 1985.
- [79] R. M. Taylor and P. J. Probert, Range finding and feature extraction by segmentation of images for mobile robot navigation, in *Proceedings of IEEE International Conference on Robotics and Automation*, Minnesota, April 1996, pp. 95–100.
- [80] J. Vandorpe, H. Van. Brussel, and H. Xu, Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2d range finder, in *Proceedings of IEEE International Conference on Robotics and Automation*, Minnesota, April 1996, pp. 901–908.
- [81] M. D. Adams. and A. Kerstens., Tracking naturally occuring indoor features in 2d and 3d with lidar range/amplitude data, *International Journal of Robotics Research*, vol. 17, pp. 907–923, 1998.
- [82] S. I. Roumeliotis and G. A. Bekey, Segments: a layered, dual kalman filter algorithm for indoor feature extraction, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Japan, April 2000, pp. 454–461.
- [83] S. B. Williams, *Efficient solutions to autonomous mapping and navigation problems*, Ph.D thesis, Sydney Univ., 2001.
- [84] D. Maksarov and H. F. Durrant-Whyte, Mobile vehicle navigation in unknown environments: a multiple hypothesis approach, *IEE Proceedings of Control Theory Application*, vol. 142, pp. 62–71, 1995.

- [85] J. Leonard and I. Cox, Modeling a dynamic environment using a bayesian multiple hypothesis approach, *Artificial Intelligence*, vol. 66, pp. 311–344, 1994.
- [86] J. E. Guivant, F. R. Masson, and E. M. Nebot, Simultaneous localization and map building using natural features and absolute information, *Robotics and Autonomous Systems*, vol. 40, pp. 79–90, 2002.
- [87] P. Dosch, G. Masini, and K. Tombre, Improving arc detection in graphics recognition, in *Proceedings of International Conference on Pattern Recognition*, Barcelona, Spain, April 2000, pp. 243–246.
- [88] D. Dori and W. Liu, Stepwise recovery of arc segmentation in complex line environments, *International Journal on Document Analysis and Recognition*, vol. 1, pp. 62–71, 1998.
- [89] R. Madhavan, H. F. Durrant-Whyte, and G. Dissanayake, Natural landmark-based autonomous navigation using curvature scale space, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, May 2002, pp. 3936–3941.
- [90] F. Tang, M. D. Adams J. Ibañez-Guzmán, and W. S. Wijesoma, Pose invariant, robust feature extraction from range data with a modified scale space approach, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, USA, Apr. 26 - May 1, 2004, pp. 3173–3179.
- [91] J. J. Leonard and H. F. Durrant-Whyte, Dynamic map building for an autonomous mobile robot, *International Journal of Robotics Research*, vol. 11, pp. 286–298, 1992.
- [92] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, Real time data association for fastslam, in *Proceedings of IEEE International Conference on Robotics and Automation*, Taiwan, September 2003, pp. 2512–2517.
- [93] T. Bailey, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte, Data association for mobile robot navigation: a graph theoretic approach, in

- Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, California, April 2000, pp. 2512–2517.
- [94] J. H. Lim and J. J. Leonard, Mobile robot relocation from echolocation constraints, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1035–1041, 2000.
- [95] J. A. Castellanos, J. M. Montiel, J. Neira, and J. D. Tardós, The spmap: a probabilistic framework for simultaneous localization and map building, *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 948–953, 1999.
- [96] Y. Bar-shalom and T. E. Fortmann, *Tracking and data association*, Academic Press, 1988.
- [97] D. B. Reid, An algorithm for tracking multiple targets, *IEEE Transactions on Automatic Control*, vol. 24, pp. 843–854, 1979.
- [98] C. L. Morefield, Application of 0-1 integer programming to multitarget tracking problems, *IEEE Transactions on Automatic Control*, vol. AC-22, No. 3, pp. 302–312, 1977.
- [99] A. B. Poore and A. J. Robertson, A new multidimensional data association algorithm for multisensor multitarget tracking, in *Proceedings of SPIE*, Vol. 2561, July 1995, pp. 448–459.
- [100] P. P. A. Storms and F. C. R. Spijksma, An lp-based algorithm for the data association problem in multitarget tracking, *Computers and Operation Research*, vol. 30, pp. 1067–1085, 2003.
- [101] J. J. Leonard and H. J. S. Feder, Decoupled stochastic mapping, *IEEE Journal of Oceanic Engineering*, vol. 26, pp. 561–571, 2001.
- [102] W. D. Rencken, Concurrent localisation and map building for mobile robots using ultrasonic sensors, in *Proceedings of IEEE Conference on Intelligent Robots and Systems*, Yokohama, Japan, October 1993, pp. 2192–2197.
- [103] G. Dissanayake, H. F. Durrant-Whyte, and T. Bailey, A computationally efficient solution to the simultaneous localisation and map building (slam)

- problem, in *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, April 2000, pp. 1009–1014.
- [104] G. Dissanayake, H. F. Durrant-Whyte, and T. Bailey, Map management for efficient simultaneous localization and mapping (slam), *Autonomous Robots*, vol. 12, pp. 267–286, 2002.
- [105] J.J. Leonard and H. Feder, A computationally efficient method for large-scale concurrent mapping and localization, in *The Ninth International Symposium on Robotics Research*, Utah, USA, 2000, pp. 316–321.
- [106] K. Chong and L. Kleeman, Feature-based mapping in real, large scale environments using an ultrasonic array, *International Journal of Robotics Research*, vol. 18, pp. 3–19, 1999.
- [107] K. Chong and L. Kleeman, Large scale sonar array mapping using multiple connected local maps, in *Field and Service, Robotics*, A. Zelinsky(Ed.), Springer-Verlag: Berlin, 1998, pp. 507–514.
- [108] P. M. Newman, *On the structure and solution of the simultaneous localization and mapping problem*, Australian Center for Field robotics, Univ. of Sydney, 1999.
- [109] M. Csorba and H. Durrant-Whyte, A new approach to map building using relative position estimates, in *Proceedings of SPIE Vol. 3087*, Orlando, USA, April 1997, pp. 115–125.
- [110] J. K. Uhlmann, *Dynamic map building and localization for autonomous vehicles*, Univ. of Oxford, 1995.
- [111] S. J. Julier and J. K. Ulmann, Simultaneous localization and map building using split covariance intersection, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Hawaii, USA, November 2001, pp. 1257–1262.
- [112] J. Guivant and E. M. Nebot, Optimization of the simultaneous localization and map building algorithm for real time implementation, *IEEE Transactions on Robotics and Automation*, vol. 17, No. 3, pp. 242–257, June 2001.

- [113] J. Guivant and E. Nebot, Solving computational and memory requirements of feature based simultaneous localization and map building algorithms, *Sydney, Australia: Australia Center for Field Robotics, Univ. Sydney*, 2002.
- [114] J. Guivant and E. Nebot, Improving computational and memory requirements of simultaneous localization and map building algorithms, in *Proceedings of IEEE International Conference on Robotics and Automation*, Washington DC, May 2002.
- [115] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, Fastslam: a factored solution to the simultaneous localization and mapping problem, in *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, Sep. 2002, pp. 67–72.
- [116] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Y. Ng, Simultaneous mapping and localization with sparse extended information filters, in *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, Nice, France, Dec. 2002, pp. 67–72.
- [117] Udo Frese and Gerd Hirzinger, Simultaneous localization and mapping - a discussion, in *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, Seattle, U.S., August 2001, pp. 67–72.
- [118] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, A solution to the simultaneous localization and map building (slam) problem, *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229–241, 2001.
- [119] S. B. Williams, G. Dissanayake, and H. F. Durrant-Whyte, An efficient solution to the simultaneous localisation and mapping problem, in *Proceedings of IEEE International Conference on Robotics and Automation*, USA, May 2002.
- [120] Z. Li and B. K. Ghosh, Geometric feature based 2D/3D map building and planning with laser, sonar and tactile sensors, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Takamatsu, Japan, October 2000, pp. 115–120.

- [121] O. A. Kai, N. Tomatis, and R. Siegwart, Multisensor on-the-fly localization using laser and vision, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Takamatsu, Japan, October 2000, pp. 462–467.
- [122] X. Zhou, Y. K. Ho, C. S. Chua, and Y. Zou, The localization of mobile robot based on laser scanner, in *Proceedings of Canadian Conference Electrical and Computer Engineering*, Canada, October 2000, pp. 841–845.
- [123] B. Blanco and B. L. Boada, Local mapping from on-line laser voronoi extraction, in *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Takamatsu, Japan, October 2000, pp. 103–108.
- [124] D. Wang, E. K. Tay, and M. Zribi, Modeling of an agv for handling heavy containers, in *Proceedings of the 3rd International Conference on Advanced Mechtronics*, Japan, August 1998, pp. 67–72.
- [125] M. D. Adams, *Sensor modelling, design and data processing for autonomous navigation*, World Sentific, 1999.
- [126] Jorge Nocedal and Stephen J. Wright, *Numerical optimization*, Springer Verlag, 1999.
- [127] L. Iocchi and D. Nardi, Hough localization for mobile robots in polygonal environments, *Robotics and Autonomous Systems*, vol. 40, pp. 43–58, 2002.
- [128] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, *IEEE Transactions on Control and Automation*, vol. 45, pp. 477–482, 2000.
- [129] J. Guivant, E. Nebot, and H. F. Durrant-Whyte, Simultaneous localization and map building using natural features in outdoor environments, in *Proceedings of IEEE International Conference on Intelligent Autonomous Systems*, Venice, Italy, November 2000, pp. 581–588.
- [130] A. B. Poore and A. J. Robertson, A new class of langrangian relaxation based algorithms for a class of multidimensional assignment problems, *Computational Optimization and Applications*, vol. 8, pp. 129–150, 1997.

- [131] A. B. Poore, Multidimensional assignment formulation of data association problems arising from multitarget tracking and multisensor data fusion, *Computational Optimization and Applications*, vol. 3, pp. 27–57, 1994.
- [132] S. Zhang, L. Xie, and M. D. Adams, An efficient data association approach to simultaneous localization and map building, in *Proceedings of IEEE International Conference on Robotics and Automation*, New Orleans, USA, Apr. 26-May 1 2004, pp. 1493–1498.
- [133] J. L. Miller and L. S. Franz, A binary rounding heuristic for multi-period variable-task duration assignment problems, *Computers and Operations Research*, vol. 23, pp. 819–828, 1996.
- [134] L. S. Franz and J. L. Miller, Scheduling medical residents to rotations: solving the large scale multiperiod staff assignment problem, *Operations Research*, vol. 41, pp. 269–279, 1993.
- [135] D. S. Hochbaum, *Approximation algorithms for NP-hard problems*, Boston: PWS Publishing Company, 1997.
- [136] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, London: The MIT Press, 2001.
- [137] R. G. Parker and R. L. Rardin, *Discrete optimization*, Academic Press, 1988.
- [138] S. Vajda, *Linear programming: algorithms and applications*, Chapman and Hall, 1981.
- [139] N. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica*, vol. 4, pp. 373–395, 1984.
- [140] J. Guivant and E. Nebot, *ACFR, Experimental outdoor dataset*, <http://www.acfr.usyd.edu.au/homepages/academic/enebot/dataset.htm>, 2003.
- [141] T. M. Cover and J. A. Thomas, *Elements of information theory*, John Wiley & Sons, 1991.

- [142] A. Doucet, N. J. Gordon, and N. de Freitas eds, *Sequential monte carlo methords in practice*, Springer Verlag, 2001.

Appendices

Appendix A

In this appendix, we shall show the vehicle's parameters and functions of its different part. The following figure shows the vehicle's construction. The names and the functions are as follows: 1. GPS receiver on the vehicle

2. Joystick for manual driving

3. Laser Measure Scanner
4. Steering actuator
5. One electric drive motor per wheel
6. One electric brake motor per wheel
7. Four lead-acid batteries and electronic management
8. Multimedia terminal
9. Infrared targets for platoon driving

Mechanics and Electronics of Cycab

The mechanics is based on a small electrical golf car frame, already produced in small series. The use of four identical wheel motor blocks, allows cost reductions (by re-using identical parts) and volume (four small engines with small power controllers are easier to build and integrate than a big one with a high power controller). Consequently, the architecture is modular and the vehicle is easier to drive (the four wheels are propulsive and directive). The steering is made through an electrical jack mechanically linked to all the wheels. Each wheel motor block has it's own power amplifier, driven by the new Motorola MPC555 microcontroller. This microcontroller is very well suited for motor control in automobiles and is capable of doing real-time data transfer. Each wheel node controls a drive motor and a brake motor, with all their associated sensors (optical encoder, brake torque measurement, temperature etc.) connected. An additional node is attached to the steering jack and the joystick.

Communications between the nodes are made through a CAN (Controller Area Network) serial bus. It has been designed specially for automotive applications and allows safe communications in disturbed environment, with a rate of up to 1Mbits/s. It carries messages of up to 8 bytes length with 50 per cent control and

arbitration bits overload. The network consists of 5 nodes and an on-board PC, which drives the screen and the hard disk.

All the 3 MPC555 microcontroller are connected to the Linux PC on board the Cycab by the CAN network.

There are 2 MPC555 microcontroller in the network that control the low-level control functions of the motor nodes in real-time. These control functions that are programmed inside the MPC555 include PID feedback loop control programs. One of the MPC555 is in charge of the front wheels and front steering while the other controls the back wheels and back steering. The 3rd MPC555 is connected to external sensor equipment like the SICK LMS in this study. It was proposed that the communication with the LMS be made via a real-time serial interface using RS422. This MPC555 is programmed to control the SICK LMS and retrieve all scan data values from it. Then, these data are filtered and only the necessary, required values are sent to the Linux PC via the CAN network. The Linux PC contains all the high-level application control programs like Localization, Trajectory Planning and Car Following. When the Linux PC receives the filtered data from the MPC555, the programs will process these vital data and generate results based on its control algorithm. Then, these results are redistributed to the front and back MPC555 for processing. Finally, these 2 MPC555 will control the motor nodes based on the results.

The onboard computer of the Cycab runs under the Albatros OS. This operating system understands the standard commands from a remote machine, the operator or an application through a serial line. The most common commands are SERV, POS, MOVE, and CPOS. For example, to communicate with the Cycab one has just to send a Cstring through the serial port. Then the Albatros OS takes charge of translating this command in an electrical signal to the motors bounded with the wheels.

Before every communication with the Cycab, it is compulsory to send a telegram

of initialization. This one advertises the Cycab that it has to be ready to react to the different commands, which will be sent to it.

The Cycab parameters are as follows:

Length	1.9 m
Width	1.2 m
Weight	300 kg
Power	4 × 1 kW Electric Motors
Drive System	4 Wheel-Drive, 4 Wheel-Steering
Maximum Speed	30 km/h
Range	40 km
Capacity	2 adults (+ luggage)
Modes of Operation	Manual driving with joystick Fully automated driving Teleoperated driving
Recharging System	Induction charging
Security	Smart card access

Laser scanner parameters

The following table shows the laser parameters:

Appendix B

The equations of EKF

Assume the system equation is as follows:

$$\mathbf{x}(k + 1) = \mathbf{f}(\mathbf{x}(k)) + \mathbf{v}(k).$$

(B-1)

Given the estimate $\hat{\mathbf{x}}(k \mid k)$ of $\mathbf{x}(k)$, the predicted state $\hat{\mathbf{x}}(k + 1 \mid k)$ using (B-1) is

Dimensions (W×H×D)	155×210×156 (mm)
Maximum Range	80 m (outdoor); 150 m (indoor)
Range (without supplementary re- flectors)	Up to 30 m
Range with minimum reflectivity 1.8%	4 m
Lateral Resolution (mm)	10 mm
Angular resolution	100° Scan: 0.25°/0.5°/1°, No. Of Scan Values: 401/201/101 180° Scan: 0.5°/1° No. Of Scan Values: 361/181 Configuration by software
Response times	52 ms with angular resolution of 0.25° 26 ms with angular resolution of 0.5° 13 ms with angular resolution of 1.0°
Interfaces	Serial RS-422 or RS-232 switchable
Measurement data	Real-time transfer, binary data
Data transfer rate	9.6 / 19.2 / 38.4 / 500 Kbaud
Voltage supply (electronic)	24 VDC ±15%
Laser Protection Class	1 (Eye-safe)

given by

$$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{f}(\hat{\mathbf{x}}(k \mid k)). \tag{B-2}$$

The prediction error covariance matrix is approximately given by:

$$\mathbf{P}(k+1 \mid k) = \mathbf{F}(k)\mathbf{P}(k \mid k)\mathbf{F}^T(k) + \mathbf{Q}(k). \tag{B-3}$$

In view of the system observation model:

$$\mathbf{z}(k) = \mathbf{H} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \mathbf{w}(k), \tag{B-4}$$

the predicted measurement is simply:

$$\hat{\mathbf{z}}(k+1) = \mathbf{H}\hat{\mathbf{x}}(k+1 \mid k). \tag{B-5}$$

Then, the difference between the measurement and the predicted observation, named the innovation, is given by:

$$\begin{aligned}\nu(k+1) &= \mathbf{z}(k+1) - \mathbf{H}\hat{\mathbf{x}}(k+1 | k) \\ &= \mathbf{H}[\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1 | k)] + \mathbf{w}(k).\end{aligned}\tag{B-6}$$

Thus, the covariance of the innovation is:

$$\mathbf{s}(k+1) = \mathbf{H}\mathbf{P}(k+1 | k)\mathbf{H}^T + \sigma_r^2.\tag{B-7}$$

The EKF gain is given by:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1 | k)\mathbf{H}^T\mathbf{s}^{-1}(k+1).\tag{B-8}$$

We update the estimation using the following equations:

$$\hat{\mathbf{x}}(k+1 | k+1) = \hat{\mathbf{x}}(k+1 | k) + \mathbf{K}(k+1)\nu(k+1),\tag{B-9}$$

$$\mathbf{P}(k+1 | k+1) = \mathbf{P}(k+1 | k) - \mathbf{K}(k+1)\mathbf{s}(k+1)\mathbf{K}^T(k+1).\tag{B-10}$$

The equations of UKF

Given the estimate $\hat{\mathbf{x}}(k | k)$ of $\mathbf{x}(k)$ and its estimation error covariance $\mathbf{P}(k | k)$, in order to avoid the linearization involved in the EKF, the UKF works by generating a set of points whose sample mean and sample covariance are $\hat{\mathbf{x}}(k | k)$ and covariance $\mathbf{P}(k | k)$, respectively. The nonlinear function is applied to each of these points in turn to yield a transformed sample, and the predicted mean and covariance are calculated from the transformed sample. The samples are deterministically chosen so that they capture specific information about the Gaussian distribution [128].

1. Form weighted samples

The n -dimensional random variable $\mathbf{x}(k)$ with mean $\hat{\mathbf{x}}(k | k)$ and covariance $\mathbf{P}(k | k)$ is approximated by $2n + 1$ weighted samples or sigma points selected by the algorithm

$$\left\{ \begin{array}{l} \chi_0(k | k) = \hat{\mathbf{x}}(k | k) \\ W_0 = \kappa / (n + \kappa) \\ \chi_i(k | k) = \hat{\mathbf{x}}(k | k) + \left(\sqrt{(n + \kappa) \mathbf{P}(k | k)} \right)_i \\ W_i = 1 / \{2(n + \kappa)\} \\ \chi_{i+n}(k | k) = \hat{\mathbf{x}}(k | k) - \left(\sqrt{(n + \kappa) \mathbf{P}(k | k)} \right)_i \\ W_{i+n} = 1 / \{2(n + \kappa)\} \end{array} \right. \quad (\text{B-11})$$

where $\kappa \in R$, $\left(\sqrt{(n + \kappa) \mathbf{P}(k | k)} \right)_i$ is the i th row or column of the matrix square root of $(n + \kappa) \mathbf{P}(k | k)$, and W_i is the weight that is associated with the i th point. In theory, κ can be any number (positive or negative) providing that $(n + \kappa) \neq 0$.

2. Prediction

Given the set of samples generated by (B-11), the prediction procedure is as follows.

a) Each sigma point is instantiated through the process model to yield a set of transformed samples

$$\chi_i(k + 1 | k) = \mathbf{f}_2[\chi_i(k | k)]. \quad (\text{B-12})$$

b) The predicted mean is computed as

$$\hat{\mathbf{x}}(k + 1 | k) = \sum_{i=0}^{2n} W_i \chi_i(k + 1 | k). \quad (\text{B-13})$$

c) The predicted covariance is computed as

$$\mathbf{P}(k + 1 | k) = \sum_{i=0}^{2n} W_i \{ \chi_i(k + 1 | k) - \hat{\mathbf{x}}(k + 1 | k) \} \{ \chi_i(k + 1 | k) - \hat{\mathbf{x}}(k + 1 | k) \}^T. \quad (\text{B-14})$$

It is also clear that the predicted measurement is simply:

$$\hat{\mathbf{z}}(k+1) = \mathbf{H}_2 \hat{\mathbf{x}}(k+1 | k). \quad (\text{B-15})$$

The difference between the measurement and the predicted observation, named the innovation, can be written as

$$\nu(k+1) = \mathbf{z}(k+1) - \mathbf{H}_2 \hat{\mathbf{x}}(k+1 | k). \quad (\text{B-16})$$

The covariance of this quantity is

$$\mathbf{s}_\nu(k+1 | k) = \mathbf{H}_2 \mathbf{P}(k+1 | k) \mathbf{H}_2^T + \sigma_r^2. \quad (\text{B-17})$$

3. Calculate the Kalman filter gain

$$\mathbf{K}(k+1) = \mathbf{P}(k+1 | k) \mathbf{H}_2^T \mathbf{s}_\nu^{-1}(k+1 | k). \quad (\text{B-18})$$

4. Update

We update the estimation using the following equations:

$$\hat{\mathbf{x}}(k+1 | k+1) = \hat{\mathbf{x}}(k+1 | k) + \mathbf{K}(k+1) \nu(k+1), \quad (\text{B-19})$$

$$\mathbf{P}(k+1 | k+1) = \mathbf{P}(k+1 | k) - \mathbf{K}(k+1) \mathbf{s}_\nu(k+1) \mathbf{K}^T(k+1). \quad (\text{B-20})$$

Appendix C

In order to simplify the presentation, we rewrite the equations (3.19) and (3.20) as follows:

$$x_{k+1} = f(x_k, u_k, v_k), \quad (\text{C-1})$$

$$z_{i,k+1} = h_i(x_{k+1}, \eta_i, \omega_{i,k}). \quad (\text{C-2})$$

The particle filtering requires to construct the PDF of the current state x_{k+1} , given all the available information, i.e. to construct $p(x_{k+1} | Z_{k+1})$, where Z_{k+1} is the set of observations up to $k+1$. In principle this PDF may be obtained recursively in two stages [142]: prediction and update. Suppose the required PDF $p(x_k | Z_k)$ is available. Then using the system model it is possible to obtain the prior PDF of the state at time step $k+1$:

$$p(x_{k+1} | Z_k) = \int p(x_{k+1} | x_k) p(x_k | Z_k) dx_k. \quad (\text{C-3})$$

Here the probability model of the state evolution $p(x_{k+1} | x_k)$, which is a Markov model, is defined by the system equation and the known statistics of v_k :

$$p(x_{k+1} | x_k) = \int \delta(x_{k+1} - f(x_k, u_k, v_k)) p(v_k) dv_k \quad (\text{C-4})$$

where $\delta(\cdot)$ is the Dirac delta function. The delta function arises because if x_k and v_k are known, then x_{k+1} is obtained from a purely deterministic relationship, see (C-1). Then at time instant $k+1$ a measurement z_{k+1} becomes available and may be used to update the prior via the Bayes rule.

$$p(x_{k+1} | Z_{k+1}) = \frac{p(z_{k+1} | x_{k+1}) p(x_{k+1} | Z_k)}{p(z_{k+1} | Z_k)} \quad (\text{C-5})$$

where the normalizing denominator is given by:

$$p(z_{k+1} | Z_k) = \int p(z_{k+1} | x_{k+1}) p(x_{k+1} | Z_k) dx_{k+1}. \quad (\text{C-6})$$

The conditional PDF of z_{k+1} given x_{k+1} is defined by the measurement model and the known statistics of v_{k+1} as follows:

$$p(z_{k+1} | x_{k+1}) = \int \delta(z_{k+1} - h_i(x_{k+1}, \omega_{i,k+1})) p(\omega_{i,k+1}) d\omega_{i,k+1}. \quad (\text{C-7})$$

In the update equation, equation (C-5), the measurement z_k is used to modify the predicted prior from the previous time step to obtain the required posterior of the state. The recurrence relations of equations (C-3) and (C-5) constitute the formal solution to the Bayesian recursive estimation problem. Based on the Bayesian estimation formula, the particle filter implementation algorithm for localization is derived as follows:

(1). Initialization: generate $x_0^{(l)} \sim p_{x_0}$, $l = 1, \dots, N$. Each sample of the state vector is referred to as a particle. That is, start from a random measure with equal weight on each of the N sample values.

(2). Prediction: predict new particles, i.e.,

$$x_{k+1}^{(l)} = f(x_k^{(l)}, u_k, v_k^{(l)}), \quad l = 1, \dots, N \quad (\text{C-8})$$

using different noise realizations of v_k for the particles. Compute the weights

$$w_{k+1}^{(l)} = p(z_{k+1} | x_{k+1}^{(l)})$$

and normalize the weight,

$$\tilde{w}_{k+1}^{(l)} = w_{k+1}^{(l)} / \sum_{j=1}^N w_{k+1}^{(j)}, \quad l = 1, \dots, N.$$

In this step, we can also get the observation particles for the i th feature:

$$z_{i,k+1}^{(l)} = h_i(x_{k+1}^{(l)}, \eta_i), \quad (\text{C-9})$$

$$\nu_{i,k+1}^{(l)} = z_{i,k+1} - z_{i,k+1}^{(l)}. \quad (\text{C-10})$$

Their mean is given by

$$\hat{\nu}_{i,k+1} = \sum_{l=1}^N \tilde{w}_{k+1}^{(l)} \nu_{i,k+1}^{(l)}. \quad (\text{C-11})$$

The variance of $\hat{\nu}_{i,k+1}$ is:

$$\hat{s}_{i,k+1} = \sum_{l=1}^N \tilde{w}_{k+1}^{(l)} (\nu_{i,k+1}^{(l)} - \hat{\nu}_{i,k+1})(\nu_{i,k+1}^{(l)} - \hat{\nu}_{i,k+1})^T. \quad (\text{C-12})$$

(3). Update: generate a new set $\{x_{k+1}^{l*}\}_{l=1}^N$ by resampling with replacement N times from $\{x_{k+1}^l\}_{l=1}^N$, where

$$Pr(x_{k+1}^{(l*)} = x_{k+1}^{(l)}) = \tilde{w}_{k+1}^{(l)} \quad l, l* = 1, \dots, N.$$

Here, the mean of $x_{k+1}^{(l*)}$ is:

$$\hat{x}_{k+1} = \sum_{l*=1}^N \frac{1}{N} x_{k+1}^{(l*)}.$$

(4). Increase k and go back to step (2).