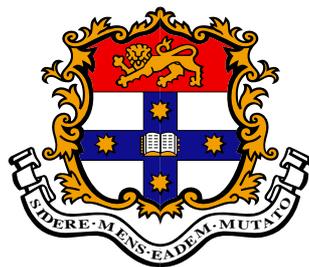


Mobile Robot Localisation and Mapping in Extensive Outdoor Environments

Tim Bailey

A thesis submitted in fulfillment
of the requirements for the degree of
Doctor of Philosophy



Australian Centre for Field Robotics
Department of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

August 2002

Declaration

This thesis is submitted to the University of Sydney in fulfillment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Tim Bailey

7 August, 2002

Reprinted with corrections and emendations 21 November, 2002

Abstract

Tim Bailey
The University of Sydney

Doctor of Philosophy
August 2002

Mobile Robot Localisation and Mapping in Extensive Outdoor Environments

This thesis addresses the issues of scale for practical implementations of *simultaneous localisation and mapping* (SLAM) in extensive outdoor environments. Building an incremental map while also using it for localisation is of prime importance for mobile robot navigation but, until recently, has been confined to small-scale, mostly indoor, environments. The critical problems for large-scale implementations are as follows. First, data association—finding correspondences between map landmarks and robot sensor measurements—becomes difficult in complex, cluttered environments, especially if the robot location is uncertain. Second, the information required to maintain a consistent map using traditional methods imposes a prohibitive computational burden as the map increases in size. And third, the mathematics for SLAM relies on assumptions of small errors and near-linearity, and these become invalid for larger maps.

In outdoor environments, the problems of scale are exacerbated by complex structure and rugged terrain. This can impede the detection of stable discrete landmarks, and can degrade the utility of motion estimates derived from wheel-encoder odometry.

This thesis presents the following contributions for large-scale SLAM. First, a batch data association method called *combined constraint data association* (CCDA) is developed, which permits robust association in cluttered environments even if the robot pose is completely unknown. Second, an alternative to feature-based data association is presented, based on correlation of unprocessed sensor data with the map, for environments that don't contain easily detectable discrete landmarks. Third, methods for feature management are presented to control the addition and removal of map landmarks, which facilitates map reliability and reduces computation. Fourth, a new map framework called *network coupled feature maps* (NCFM) is introduced, where the world is divided into a graph of connected submaps. This map framework is shown to solve the problems of consistency and tractability for very large-scale SLAM.

The theoretical contributions of this thesis are demonstrated with a series of practical implementations using a scanning range laser in three different outdoor environments. These include: sensor-based dead reckoning, which is a highly accurate alternative to odometry for rough terrain; correlation-based localisation using particle filter methods; and NCFM SLAM over a region greater than 50000 square metres, and including trajectories with large loops.

Acknowledgements

This work has benefited greatly from the input and support of many people over the past five years, and my gratitude at its completion is extended especially to those mentioned below.

To begin, I would like to thank my supervisors. To Eduardo Nebot, thanks for your availability, expertise and advice, motivation, and encouragement to move forward. To Julio Rosenblatt, thanks for your alternative perspective, in particular for introducing me to the concept of topological maps. Also, Hugh Durrant-Whyte, while not an official supervisor, deserves special thanks for his reviews of this thesis, helpful comments and wise counsel.

Thank you to Sebastian Thrun for his review of this thesis, for his suggestions and encouragement. His advice is greatly appreciated.

The team at the ACFR have made my time enjoyable and rewarding, and have often provided a welcome respite from research. In particular, I must thank Monica Louda and Mike Stevens for their valued friendship. To Monica, thanks for being such a long-suffering neighbour and for all those cups of coffee. To Mike, thanks for your intelligence, humility and humour—you are an inspiration. I am indebted to Stefan Williams both for his excellent thesis and for many stimulating conversations, thank you. To Jose Guivant and Som Majumder, thanks for the willingness with which you shared your knowledge and research. Thanks also to Jose and the ute team for the park data set, and to Eric Nettleton for the mine data set. And thanks to “Dissa” Dissanayake for his interest and persistence in asking me “have you finished yet.” To the remaining members of the ACFR, notably Steve Scheduling, Keith Willis, Ben Grocholsky, and Ralph Koch, thanks for your company and for many entertaining lunchtime discussions.

My most valuable support by far has come from my lovely wife Angela. Thank you, Ang, for your constancy in love, support and understanding. To Mum and Dad, thank you for a lifetime of love and care, for your prayers, and for financing my life and education all those years.

Finally, I thank my Lord Jesus who supplies all my needs. Thank you for inspiration and perseverance. Thank you, above all, for the incomparable gift of salvation and the eternal hope you offer all who would place their trust in you.

There is a fine line between thorough and slow.

Anon.

We know that we all possess knowledge. Knowledge puffs up, but love builds up. The man who thinks he knows something does not yet know as he ought to know. But the man who loves God is known by God.

1 Corinthians 8:1b-3

Contents

Abstract	ii
Acknowledgements	iii
Contents	v
1 Introduction	1
1.1 Mobile Robot Localisation and Mapping	2
1.1.1 Forms of Localisation	2
1.1.2 A Brief History of Autonomous Localisation	4
1.2 Contributions	5
1.3 Thesis Overview	6
2 Autonomous Localisation	8
2.1 Navigational Maps and Their Application to SLAM	8
2.1.1 Occupancy Grids	10
2.1.2 Feature Maps	11
2.1.3 Topological Maps	14
2.1.4 Hybrid Topological-Metric Maps	17
2.2 The Stochastic SLAM Algorithm	18
2.2.1 Vehicle, Map and Augmented State Vectors	18
2.2.2 Prediction Stage	19
2.2.3 Update Stage	21
2.2.4 State Augmentation	22
2.2.5 Efficiency Improvements for Large-Scale SLAM	23
2.3 Experiments in Outdoor SLAM	24
2.3.1 SLAM with a Scanning Range Laser	25
2.3.2 Two Dimensional Projection of Non-Planar Environments	26
2.3.3 Dead Reckoning in Rough Terrain	26
2.3.4 Ramifications of GPS	28
2.4 Summary	29

3	Batch Data Association for Correlated Feature Sets	30
3.1	Data Association For Target Tracking	31
3.1.1	Validation Gating	31
3.1.2	Tracking a Single Target in Clutter	34
3.1.3	Tracking Multiple Targets in Clutter	34
3.2	Batch Data Association For Correlated Feature Sets	36
3.2.1	Joint Compatibility Branch and Bound	37
3.2.2	Maximum Common Subgraph	39
3.3	Combined Constraint Data Association	41
3.3.1	Calculation of Relative and Absolute Constraints	42
3.3.2	Set and Graph Terminology	44
3.3.3	Correspondence Graph with Unknown Vehicle Pose	45
3.3.4	Correspondence Graph with Partially Known Vehicle Pose	49
3.3.5	Maximum Clique Search	53
3.3.6	Computational Complexity of the Maximum Clique Search	55
3.3.7	Ambiguity Management	58
3.3.8	A Comparison of Optimal Assignment, JCBB and CCDA	59
3.4	Application: Sensor-based Dead Reckoning	60
3.4.1	Feature Extraction	60
3.4.2	Dead Reckoning Implementation Details	64
3.4.3	Results	64
3.5	Remarks	65
3.5.1	Utility of Simple Geometric Features	65
3.5.2	Minimum Acceptable Batch Size	68
3.6	Summary	68
4	An Alternative to Feature-based Data Association: Scan Correlation	70
4.1	A Review of Point Data Correlation Techniques	72
4.1.1	Iterative Closest Point	72
4.1.2	Angle Histogram	73
4.1.3	Occupancy Grid Correlation	74
4.1.4	Probabilistic Methods	74
4.2	A Probabilistic Representation for Unprocessed Data	75
4.2.1	Representation as a Two-Dimensional PDF	76
4.2.2	Higher Dimensional Alternatives	77
4.2.3	Application of the 2-D Representation	78
4.3	A Bayesian Likelihood Function for a Point Target Model	79
4.3.1	Notation	79
4.3.2	Robot Localisation in a Plane	79
4.4	Sum of Gaussians Scan Correlation	82

4.4.1	Conversion to Gaussian Sum Representation	82
4.4.2	Scaling Models for Gaussian Sum	83
4.4.3	Cross-Correlation in One Dimension	86
4.4.4	Planar Cross-Correlation	87
4.4.5	An Alternative Correlation Scheme	88
4.4.6	A Comparison of Correlation and Explicit Data Association	89
4.5	Application: Maximum Likelihood Dead Reckoning	93
4.5.1	Results	93
4.5.2	Multimodal Likelihood Functions	96
4.5.3	Discussion of Results	99
4.6	Application: Particle Filter Localisation	99
4.6.1	State Vector and Process Model	99
4.6.2	Likelihood Modification	100
4.6.3	Results	101
4.7	Remarks: A 2-D PDF for Map Building and SLAM	104
4.8	Summary	105
5	Considerations for SLAM in Moderate-Sized Environments	106
5.1	Feature Management	106
5.1.1	Feature Addition	107
5.1.2	Constrained Initialisation	107
5.1.3	Density Control	109
5.1.4	Obsolete Feature Removal	111
5.2	Application: Partial SLAM	113
5.2.1	Results	115
5.2.2	Partial SLAM and GPS	115
5.3	Data Association and the Cycle Detection Problem	118
5.4	Application: Full SLAM	118
5.4.1	Results With Medium-Scale Loop Closure	120
5.4.2	Results With Large-Scale Loop Closure	123
5.5	Remarks: SLAM versus Batch Map Building	125
5.6	Summary	125
6	Network Coupled Feature Maps	127
6.1	Submap Methods for Stochastic SLAM	128
6.1.1	Decoupled Stochastic Mapping	128
6.1.2	Two-Level Landmark Representation	129
6.1.3	Relative Landmark Representation	130
6.1.4	Hierarchical Local Maps	131
6.1.5	Constrained Relative Submap Filter	133

6.2	Network Coupled Feature Maps	134
6.2.1	NCFM Overview	135
6.2.2	Coupling Estimate Equations	136
6.2.3	Map Traversal	139
6.2.4	Submap Creation	140
6.2.5	Coupling Convergence	141
6.2.6	Consequences of Non-independent Couplings	144
6.3	Cycle Detection within the NCFM Framework	147
6.3.1	First Pass Detection	147
6.3.2	Second Pass Detection	151
6.3.3	Cycle Confirmation	152
6.3.4	Pathological Symmetries	152
6.4	Application: Large-scale Outdoor NCFM SLAM	153
6.4.1	Implementation Details	153
6.4.2	Results	154
6.5	Summary	157
7	Conclusion	158
7.1	Summary of Contributions	158
7.1.1	Combined Constraint Data Association	158
7.1.2	Sum of Gaussians Scan Correlation	159
7.1.3	Feature Management	159
7.1.4	Network Coupled Feature Maps	160
7.2	Future Research	160
7.2.1	CCDA Algorithm Extensions	161
7.2.2	Scan Correlation Applications	161
7.2.3	Feature Management Extensions	162
7.2.4	NCFM SLAM Extensions	162
7.2.5	Longer Term Developments	162
A	Environments	164
A.1	Urban Parkland	164
A.2	Internal Road of a Country Resort	167
A.3	Underground Mine Tunnel	169
B	Kalman Filter Topics	171
B.1	State-Space	171
B.1.1	Linear Transformations	172
B.2	The Kalman Filter	173
B.3	The Extended Kalman Filter	174

C	Insert and Observe State Augmentation	176
C.1	New Feature Initialisation	176
C.2	Vehicle Pose Initialisation in a Partially Known Map	177
C.3	Relative Pose Estimation	178
D	Particle Filters	181
D.1	Probability Density Functions	181
D.2	Recursive Bayesian Estimation	182
D.3	Particle Filtering: Sample Based Estimation	183
D.4	Resampling Implementations	186
D.5	Deficiencies of the Particle Filter Algorithm	188
E	Volume from Gaussian Multiplication	189
E.1	Multiplication of Two Gaussians	189
E.2	Volume from the Multiplication of Two Gaussians	191
	Bibliography	193

Chapter 1

Introduction

Autonomous mobile robot localisation in previously unexplored environments requires the robot to incrementally construct a map of its surroundings by a process called *simultaneous localisation and mapping* (SLAM). This thesis is concerned with the implementation of SLAM in extensive environments that possess difficult properties like rough terrain and lack of man-made structure. The development of a feasible and reliable SLAM system in these types of environments is dependent on addressing the following issues.

- **Data association.** SLAM relies on correct correspondence between data obtained from the robot sensors and the data currently stored in the map. In natural environments the varied distribution of recognisable objects (features) necessitates a data association algorithm robust to both high feature density and an absence of stable features. There may also be a high proportion of dynamic objects and spurious sensor measurements, and these difficulties are further accentuated if the robot location is unknown or highly uncertain.
- **Computation.** For real-time operation in very large environments, the computational complexity and storage requirements of the SLAM algorithm must scale in a reasonable manner.
- **Non-linearity.** The process of constructing an incremental map while also localising from it is highly non-linear. In large environments involving significant error accumulation and correction, the SLAM algorithm must remain mathematically consistent.
- **Adaptation.** In the presence of dynamic objects and changing environmental structure, it is necessary to establish criterion for information removal from the map so that it maintains a contemporary representation of the real world.

The objective of this thesis is to develop solutions to the issues specified above and demonstrate the functionality of these solutions with experimentation in real outdoor environments.

1.1 Mobile Robot Localisation and Mapping

The ability of a mobile robot to determine its location in space is a fundamental competence for autonomous navigation. Knowledge of self-location, and the location of other places of interest in the world, is the basic foundation on which all high level navigation operations are built. It enables strategic path planning for tasks such as goal reaching, region coverage, exploration and obstacle avoidance, and makes following of these planned trajectories possible. Without a notion of location, a robot is limited to reactive behaviour based solely on local stimuli and is incapable of planning actions beyond its immediate sensing range.

1.1.1 Forms of Localisation

This section describes three types of localisation—dead reckoning, *a priori* map localisation, and SLAM—representing increasing levels of competence in pose estimation.¹

The most basic form of localisation is dead reckoning, which is simply estimation of the vehicle pose by integrating estimates of its motion (e.g., inertial sensing, encoder-based odometry). The problem with dead reckoning is that each change-in-pose estimate includes a component of error and these errors accumulate as part of the integration process. Thus, uncertainty in the pose estimate increases monotonically with time and improving sensor and motion model accuracy can serve only to slow, but not prevent, this increase. Eventually, the pose estimate must become so uncertain that it can serve no useful purpose and, for this reason, dead reckoning is an insufficient mechanism for long-term localisation. Dead reckoning does, however, retain usefulness as an auxiliary information source in conjunction with map-based localisation.

Pose estimation with bounded uncertainty is only possible through the availability of *absolute* rather than incremental pose measurements. A map of the environment defined by the locations of distinct landmarks provides such a source of absolute information. Thus, given an ability to sense its surroundings, the robot can obtain absolute pose estimates by registering sensed information with the map. The problem with *a priori* map based localisation is the need to have explored the environment in advance, and to have surveyed the landmark locations before the robot can begin to navigate autonomously. Construction of an *a priori* map may be a difficult operation and a new map must be built for each new environment. Moreover, the resulting map is static and cannot adapt to changes in the environment or grow with exploration into regions beyond the original map bounds.

The motivation for SLAM is to overcome the need for *a priori* maps as a mechanism for bounded pose uncertainty, and to enable map construction that is extensible and adaptive to environmental change. SLAM is performed by storing landmarks in a map as they are observed by the robot sensors, using the robot pose estimate to determine the landmark locations, while at the same time, using these landmarks to improve the robot pose estimate (see Figure 1.1). As the landmarks are repeatedly reobserved, their locations become increasingly certain and the map converges, eventually acquiring the rigidity of an *a priori* map.

¹This thesis investigates the localisation problem in the context of 2-D (planar) environments, meaning that the location of the robot is given by its *pose* (i.e., position (x, y) and orientation ϕ). However, the concepts and results presented are equally valid for 3-D environments.

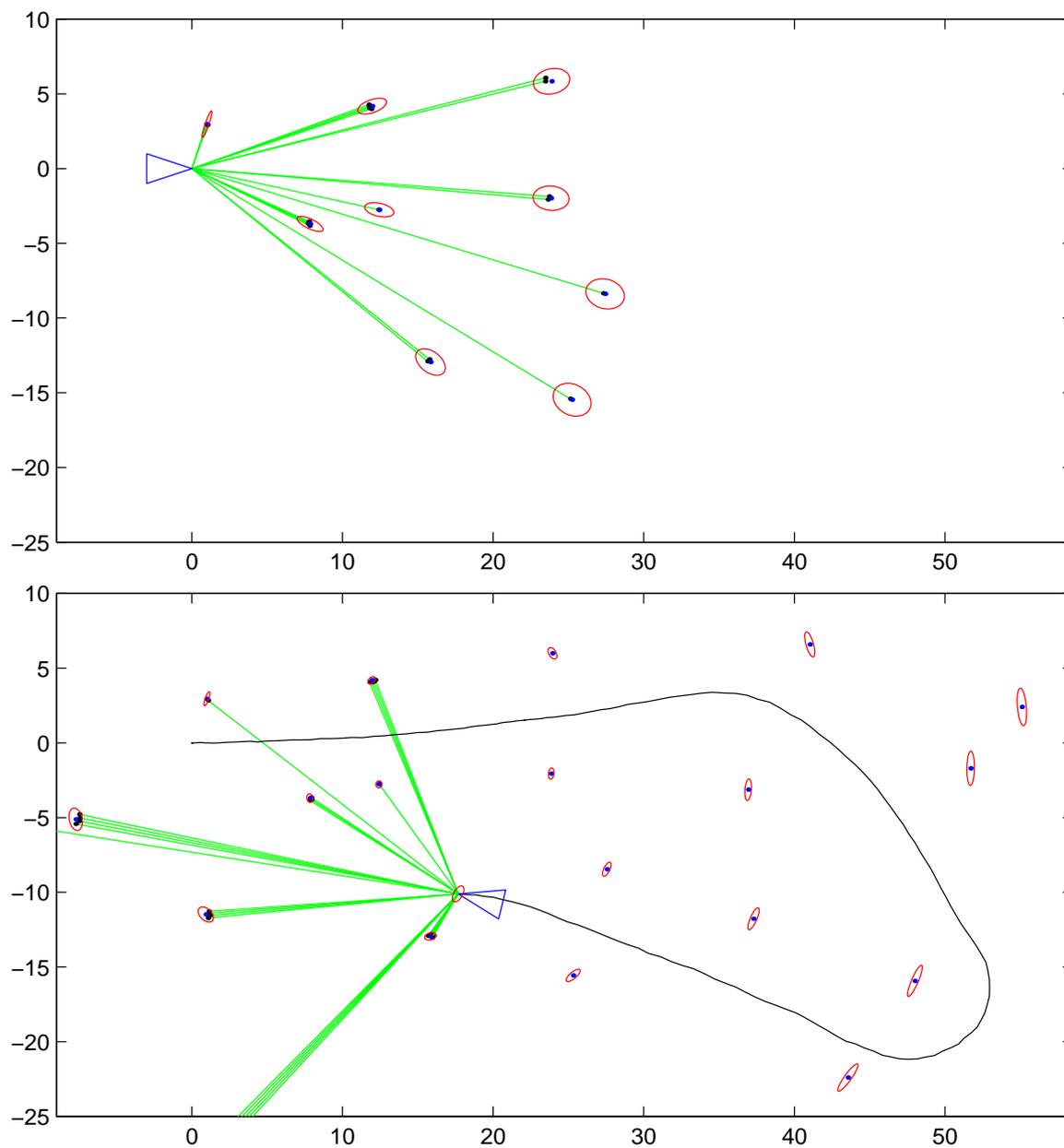


Figure 1.1: A SLAM example. The vehicle (triangle) obtains a set of range-bearing measurements (green rays). From these measurements, landmarks are extracted; new landmarks are added to the map and reobserved landmarks are used to improve the estimates of the vehicle pose and known landmark positions. The landmark uncertainties are shown by red ellipses (using a 5σ bound). Note, the uncertainty of the vehicle and added landmarks increases with distance from the map origin, but the uncertainty of each landmark decreases *monotonically* once initialised. In the limit, the landmark uncertainties approach a lower bound and their uncertainty with respect to each other approaches zero [42].

The SLAM ideal is to allow immediate navigation capabilities in completely unknown environments, so that a robot might be placed in a new environment and left to explore and map the environment without human intervention. Alternatively a human operator might drive the robot through an environment once (a training phase where the operator might demonstrate trajectories and way-points) and it could accumulate sufficient knowledge to thereafter travel competently within the region. It is worth noting that, for real environments, training by a human operator is currently a more practicable scenario than autonomous exploration, as exploration is unlikely to operate safely in the presence of hidden structures (i.e., structures invisible to the robot’s sensors). For example, transparent objects, and objects outside of the view plane, are invisible to a robot with only a range laser sensor and will not be avoided. Also, structures such as stair-wells present a danger unless there is a specific recognition of vertical free-space.

1.1.2 A Brief History of Autonomous Localisation

The origin of mobile robot localisation extends back to the 1950s with the installation of wire guided tractors in industrial factories [47]. By the 1970s this path following concept had been developed to the point where *autonomous guided vehicles* (AGVs) navigated by following lines on the ground—either buried wires (via magnetic inductance) or painted stripes [133]. Buried wires were reliable and permanent but suffered from substantial installation effort and subsequent inflexibility. Painted lines enabled more rapid path generation and alteration but required continued maintenance to ensure reliability (against wear and fading). The basic limitation of path following is that it restricts navigation to fixed trajectories and, therefore, limits AGV application to simple repetitive tasks. Thus, the path following method, while not actually localisation in the pose estimation sense,² was a precursor to autonomous localisation in establishing the problem of autonomous navigation and precipitating the need for more flexible navigation strategies.

Increased flexibility via pose estimation was introduced through the use of artificial beacons. These were either active beacons such as infrared [57] or ultrasonic [80] transducers, or passive beacons such as retro-reflective markers [20] or radar trihedrals [49], and they enabled mobile robots to localise relative to the known beacon locations. This meant that the prescribed navigation paths could easily be redefined in software without any change to the physical environment and the robot could generate adaptive trajectories to bypass obstacles. Nevertheless, this method still requires the introduction of specialised infrastructure (the beacons themselves) that need to be carefully surveyed so as to provide accurate landmark locations.

The use of the natural environment structure to provide landmarks was the next step in the development of autonomous localisation, removing the need for specialised infrastructure. By providing the robot with accurate metric maps of the environment (constructed by hand or, commonly, based on CAD structural plans [33]) the sensed environment could be registered with the map to determine its location. This use of natural landmarks introduced the problem of data association—the process of finding a correspondence between elements of two data sets. In the case of mobile robot localisation, data association concerns

²Path following is a more direct forerunner to the topological map localisation paradigm [85], where a qualitative network of paths and places is employed in preference to a metric map of landmark locations.

assigning sensed features to appropriate map landmarks. While data association is also an issue in artificial landmark localisation, it can be easily avoided by keeping the beacons well separated or making each beacon uniquely distinguishable, using bar-codes for example.

Manually mapping (surveying) an environment, when CAD plans are not available, is tedious work and susceptible to human errors and inaccuracies. A better method for *a priori* map construction was demonstrated recently [131] where the robot was driven manually through the environment gathering information with its sensors. This information was subsequently used to automatically generate an environment map using an offline batch process, reducing map building from several days to a matter of hours.

The first SLAM algorithm to comprise an explicit and consistent representation of uncertainty, and therefore provide qualification of map convergence, was presented in [122]. This method, referred to here as *stochastic* SLAM, remains the basic foundation of practically all subsequent SLAM proposals using the landmark-based map framework. Early experimental verification of the algorithm with laser [102] and sonar [87] sensors demonstrated its utility in relatively small-scale indoor environments. However, for larger regions, the $O(n^2)$ computation and storage involved in building a map of n features made direct application of the algorithm intractable. More feasible adaptations of stochastic SLAM have since been proposed such as removing redundant map features [40], developing efficient estimation procedures [65, 140], and dividing the environment into a network of submaps [90, 140, 30].

For practical SLAM in large-scale unstructured environments, several open problems remain including issues regarding landmark representation, data association and map structure. First, nearly all current stochastic SLAM implementations represent landmarks as simple geometric primitives like point-location or line features and so are viable only in environments where these forms exist. In many real environments, SLAM depends on the development of more general feature models. Second, SLAM tends to fail if data association is performed incorrectly. Therefore, data association must be made robust to variable densities of (often identical) features and dynamic objects, and large uncertainties in vehicle pose. Finally, the structure of the SLAM map needs to enable consistent uncertainty representation regardless of the size of the map. This is partly a computational problem, where correlations between uncertain variables need to be maintained, but is also a mathematical issue as uncertainty must be dealt with properly in a highly non-linear system.

1.2 Contributions

This thesis is concerned with the robustness and tractability issues for practical stochastic SLAM in large-scale, particularly outdoor, environments. Specific contributions are made towards reliable data association, map management, feasible computation, and mathematical consistency. The principal contributions of this thesis are as follows:

- The development of a batch data association algorithm for sets of parametric features. This method, called *Combined Constraint Data Association* (CCDA), incorporates all available constraint information to produce an optimal set of assignments between two feature sets. Of particular value to robot navigation is its ability to perform data association with or without the availability of vehicle pose information.

- The presentation of a probabilistic (Gaussian sum) representation for unprocessed sensor data, which provides a means for finding a Bayesian correlation between unprocessed data sets without feature models. This is a necessary alternative to feature-based data association in environments not suitable to geometric feature extraction.
- The investigation of feature management techniques to limit the addition of unreliable features, remove obsolete features and control feature density. These methods are vital for the adaptability and efficiency of long-term SLAM.
- The development of a hybrid topological-metric map representation, called *Network Coupled Feature Maps* (NCFM), that enables stochastic SLAM in very large-scale environments by addressing issues of computation, storage and mathematical consistency. In conjunction with the CCDA algorithm, this framework also provides a robust solution to the difficult problem of *loop closure*.
- The implementation of experiments in outdoor environments to demonstrate the utility of CCDA, Bayesian correlation, feature management, and NCFM. These applications include sensor-based (odometry-free) dead reckoning for use in rough-terrain environments, particle filter localisation, traditional stochastic SLAM, and NCFM SLAM with large-scale loop closure.

1.3 Thesis Overview

Chapter 2 presents the necessary background to this thesis by discussing common alternative map representations and their pros and cons when used for SLAM. The traditional stochastic SLAM algorithm for feature-based maps is then presented, and a discussion of the experimental issues for performing large-scale SLAM in outdoor environments.

Chapter 3 gives a detailed background to data association for multiple target tracking, and its relevance to stochastic SLAM. The CCDA algorithm is presented as a highly robust batch association method, and is then used to implement laser-based dead reckoning in rugged outdoor terrain.

Chapter 4 discusses scan correlation—the association (or alignment) of unprocessed data without using geometric feature models. A probabilistic (Gaussian sum) representation is developed for range-bearing data, which permits appropriate sensor modelling and Bayesian pose estimation. The validity of this approach is demonstrated through two applications: sensor-based dead reckoning and particle filter localisation.

Chapter 5 addresses two issues concerning long-term SLAM in medium-scale environments. The first is feature management—feature addition and removal—for efficient and reliable maps that can adapt to structural change. The second concerns loop closure, which cannot be performed robustly with batch association alone and requires feature grouping within the map. An implementation of traditional SLAM in a high feature density environment is used to verify these solutions.

Chapter 6 examines the current state-of-the-art in submap SLAM techniques, and presents the NCFM approach as a complete SLAM framework that is consistent and highly accurate at a global level. NCFM is also shown to permit very robust loop closure. Preliminary experimental NCFM SLAM results are demonstrated.

Chapter 7 presents conclusions and suggests future directions for the completion and extension of this work.

Chapter 2

Autonomous Localisation

The problem of autonomous localisation has received considerable attention over the past two decades and, as a result, a variety of paradigms exist for determining the position and orientation of a robot vehicle in relation to other objects in the environment. One attribute all localisation methods have in common is a map—a representation of the external environment that provides a reference for the information obtained from the robot sensors. This chapter examines several map forms currently used in mobile robot navigation systems. Their suitability for reliable localisation is discussed, particularly with regard to large-scale SLAM, and this evaluation leads to the selection of metric feature-based maps as a best choice for most situations. The remainder of this thesis is based on a feature map representation, although the developments in Chapter 6 draw from topological map concepts.

The later sections of this chapter provide a context for the research in this thesis by describing the following aspects of feature-based SLAM.

- The basic stochastic SLAM algorithm is presented which stores the vehicle pose and map feature parameters in a state vector and updates these estimates using an *extended Kalman filter* (EKF).
- Recent improvements in the efficiency of the SLAM algorithm are reviewed in terms of their ability to permit feasible operation in large-scale environments.
- The issues involved in performing practical SLAM experiments in rugged outdoor environments are discussed with emphasis on the particular case where a scanning range laser is the sole information source.

2.1 Navigational Maps and Their Application to SLAM

This section considers the three types of maps most commonly used in current localisation systems: occupancy grids, feature maps and topological maps. For each of these maps, a basic functional description is given and a survey is made of their suitability to *a priori* map localisation (in terms of computational complexity, reliability, etc). An examination of suitability for SLAM follows, with attention to criteria considered essential for the development

of a tractable and consistent SLAM algorithm. These criteria are given below (note, the first two criteria are valid only for metric maps).

- Representation of uncertainty. Mobile robot sensors cannot measure locations in the environment with total accuracy and so there is a degree of uncertainty in the map representation. At the same time, the vehicle location is derived from this map and so the pose estimate is also uncertain. This thesis stipulates that both map and pose estimates require a proper quantification of uncertainty, including conditional uncertainty where various factors are dependent. Essentially, an uncertainty model needs to accurately reflect the error between the estimated and actual state of the system.
- Monotonic convergence. The primary purpose of an uncertainty measure is to ensure map convergence. A map is convergent if the estimated spatial geometry of the environment approaches its true (physical) geometry as new observation information is incorporated. This is synonymous with a monotonic reduction in map uncertainty (i.e., non-increasing uncertainty of the known map). Without this uncertainty measure, a static object with estimated location (x_1, y_1) may drift with subsequent map updates to some arbitrarily distant location (x_2, y_2) . Explicit uncertainty permits formal assessment of map accuracy and constrains the effect of subsequent observation information.
- Data association. The map representation must permit reliable correspondence between the information obtained from the robot sensors and the stored map information. First, the search for observation-to-map association needs to be efficient enough for real-time operation and, second, the association must be robust to partial views and large search-spaces. Partial views occur because an observation may consist of a combination of currently mapped region, unexplored region and dynamic objects. The size of the search space is determined by the vehicle pose uncertainty; thus, an accurate uncertainty model improves both efficiency and robustness by prescribing a minimal search space.
- Cycle detection. If a mobile robot explores an environment by traversing a large loop (i.e., much larger than its sensing range), then identifying a return to an old map region is the cycle detection problem (also known as the map revisitation or loop closure problem). Cycle detection involves special-case conditions concerning the two previous criteria. The first issue is data association, which is distinct from local association because of the much larger vehicle pose uncertainty, and hence search-space, involved. Search efficiency is one aspect but, more importantly, there needs to be robustness in deciding whether an association is correct or an artifact of environmental similarity. The second issue, having found a correct association, is convergence, where substantial accumulated error in the map loop must be compensated properly during the map update phase by propagating the error-offset back through the map circuit.
- Computation and storage. The map must store sufficient information to enable data association and convergence. This storage, and the computation required to update the map with new observation data, should scale reasonably with the area of environment covered.



Figure 2.1: Occupancy grid map. The probability of occupancy for each grid square is defined by a value bounded by $(0, 1)$ such that 0 indicates definitely not occupied (free space) and 1 means definitely occupied. A prior probability of 0.5 implies unexplored space (depicted by the light gray regions). Note that the rectangular grid is not an efficient representation of non-rectangular environments such as this one.

2.1.1 Occupancy Grids

Occupancy (or evidence) grids [50, 118] represent a region as a matrix of cells as shown in Figure 2.1. Each cell describes a small rectangular area in the environment, and indicates the probability that the area is occupied by a value in the range $(0, 1)$. Localisation is accomplished by registering observation data with the map using cross-correlation methods (the same technique as used for image-based template matching).

As an *a priori* map, in reasonably small environments, occupancy grids are an effective localisation method. Typically observation data is accumulated in a short-term occupancy grid before being registered with the main map [100, 50, 142, 128]. This enables data fusion in a uniform occupancy representation—to filter noisy sensor data over a short time period, or to combine data from multiple sensors and different sensor modalities. Occupancy grids also offer an explicit representation of both occupied and free space, which is useful for path planning.

One difficulty concerning occupancy grids is data association. The cross-correlation search, within the region of the vehicle pose uncertainty, is expensive if the search-space is large (although fast search methods have been presented e.g., [82]). Also, if the cross-correlation result is multi-modal within the search-space region, a maximum likelihood correlation search may fail by converging to the wrong mode. This problem might be addressed by employing a Monte Carlo localisation procedure, as introduced in [38].

The most significant difficulty with occupancy grids, as *a priori* maps in large environments, is the tradeoff between grid resolution (termed granularity) and computational

complexity. Ideally, to capture environmental detail and to facilitate accurate pose estimation, the grid size is as small as possible (fine grained), whereas for feasible computation, given that storage and computation increases in proportion to the number of grid cells, a larger grid size may be necessary (coarse grained). Also, tasks like path-planning become computationally expensive for fine grid resolutions. Methods to obtain variable granularity, and so focus resources at regions of environmental complexity, have been presented [103, 21], but these possess implementation difficulties of their own.

Occupancy grid SLAM [142, 119] interleaves the steps of localisation and map update by first registering the short-term map with the global map (localisation), and then updating the perceived occupancy of the global map grid cells (map building). This method has been shown to work robustly in dynamic indoor environments over a limited period of time. However, as a SLAM map representation, occupancy grids do not possess an appropriate uncertainty model and so will tend to diverge in the longer term.

Occupancy grids are reasonably able to incorporate models of sensor uncertainty (e.g., see [99, 50]), and so are suitable for either localisation given an *a priori* map or map building given location. SLAM, however, requires an integrated representation of sensor and vehicle pose uncertainty and their correlations, but this is not supported within the occupancy grid framework. That is, occupancy grids can represent uncertainty at a local (vehicle-centric) level, but not at a global level—which is essential for map convergence. By not defining criteria for convergence, the developing map is able to drift with each observation update and this divergence exhibits itself as a slow blurring of the map.

Cycle detection provides the main instance for examining the failings of occupancy grid SLAM. First, without a reasonable estimate of global map uncertainty, the search-space for cycle detection is undefined and must either cover the whole space or be limited by *ad hoc* bounds. Second, for large cycles, the minimal search-space may become too great for real-time cross-correlation. Third, given a large search-space, the possibility of multiple correlation modes is high, and data association failure becomes increasingly likely. Finally, and most importantly, even if a correct correspondence between the observed data and the old map region is made, there is still no mechanism for offsetting the error accumulated in the map loop, and any large error in a map cycle will certainly result in an inconsistent map.

2.1.2 Feature Maps

Feature maps (or landmark maps) [123, 89] represent the environment by the global locations of parametric features (such as points and lines) as shown in Figure 2.2. Localisation is performed by extracting features from sensed data and associating them to features in the map. The differences between the predicted feature locations and the measured locations are then used to calculate the vehicle pose. In this way, localisation is very like a multiple target tracking problem [14] but, unlike normal target tracking, the targets are static and the *observer* is in motion.

The landmark locations in an *a priori* feature map are assumed to be perfectly known and so each feature is entirely defined by its parameter set. For example, a point-location feature for a cylindrical landmark, such as a pole or tree trunk, might be defined by $\mathbf{f} = (t, r, x, y)$, where t is the feature type (cylinder type), r is the cylinder radius, and

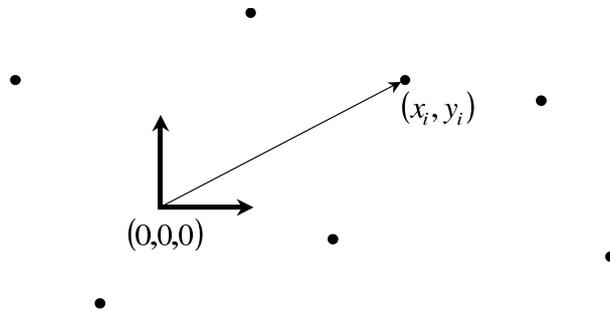


Figure 2.2: Feature map. The environment is defined by parameterised features (point locations in this example). These static landmarks are tracked using target tracking methods to determine the motion of the observer.

x and y define its centre location. Only the location information is directly useful for localisation but the other information serves to assist landmark recognition for data association. Since each landmark is represented by a limited set of parameters, the feature map $\{\mathbf{f}_1, \dots, \mathbf{f}_n\}$ is a very efficient environment representation. Unlike occupancy grids, where a dense environmental description is maintained, feature maps form a sparse representation of select landmarks. In particular, free-space is not represented and does not incur any cost in the localisation process. For this reason, feature maps do not facilitate path-planning or obstacle avoidance, and these must be performed as separate operations.

Localisation using a feature map is a parameter estimation problem to determine the vehicle pose (x, y, ϕ) given the map feature information and a set of feature observations. Assuming the measurements are correctly associated to the appropriate map features, the vehicle pose can be tracked using standard estimation techniques—with the EKF being the most common method applied to this problem [88, 49, 42].¹ Recursive EKF pose estimation has the advantages of efficient data fusion from multiple sensor measurements and the ability to incorporate explicit sensor uncertainty models.

Data association is arguably the main weakness of feature map localisation. Correct pose estimation relies on finding correct correspondence between a feature observation and its associated map feature. A misassociation results in an inconsistency where the vehicle pose uncertainty decreases but the estimate error actually increases. The effect of significant false associations is to dramatically increase the pose estimate error and prevent any subsequent map registration—so that the vehicle becomes lost. Most feature map localisation implementations are susceptible to data association failure because they rely on the association methods developed for target tracking, which treat each measurement in isolation. By failing to exploit the correlations between fixed landmarks, this approach is sensitive to observer pose uncertainty and high feature density. A dramatic increase in robustness is possible using batch data association, where a set of observations are assigned at once, as this enables association distinction based on their combined association likelihoods, effectively utilising the geometric character of the local region.

¹Details regarding state-space methods, the Kalman filter and the extended Kalman filter are provided in Appendix B.

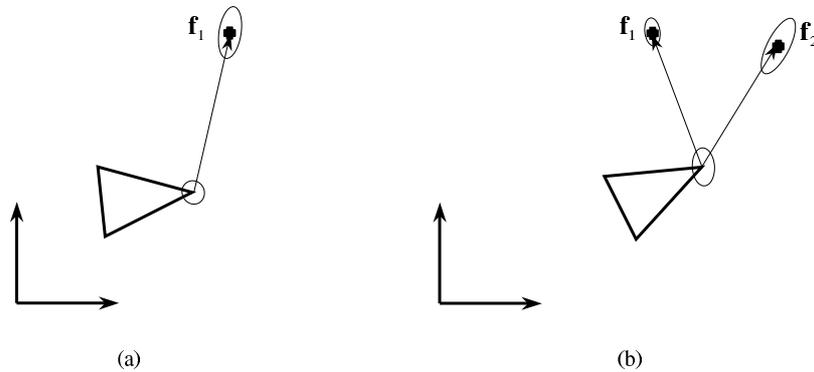


Figure 2.3: SLAM uncertainty correlations. In (a) the vehicle observes and initialises feature f_1 and so the estimate of f_1 is dependent on the vehicle pose estimate. In the next time step (b) the vehicle reobserves f_1 and initialises f_2 . Thus the estimate of f_2 is dependent on the vehicle pose which in turn is dependent on f_1 . As the features are repeatedly reobserved, they become increasingly correlated to each other and, in the limit, form a completely rigid map.

A further problem concerning feature maps is their suitability only to environments where the observed objects can be reasonably depicted by basic geometric feature models. This is often not the case in very unstructured environments where the observed objects might appear as arbitrary curves rather than, say, distinct points or lines. For reliable operation in these environments, it is necessary to devise parametric feature models that describe these general objects sufficiently well for consistent extraction and classification.

Feature map SLAM [122, 42] comprises the dual task of adding observed features to the map, using the vehicle pose as a reference, while using existing map features to estimate the vehicle pose. The uncertainty of sensor measurements, therefore, results in uncertain estimates of both the vehicle pose and the map feature locations, and these uncertainties are dependent (or correlated), as shown in Figure 2.3. Correlated uncertainty has an important consequence for feature-based SLAM as it inextricably couples the individual features to each other and the vehicle to the map. Attempts to estimate the vehicle pose and map features independently have been shown to produce inconsistent (optimistic) uncertainty estimates [87, 27].

Consistent stochastic estimation requires that correlations between parameters are explicitly maintained. For an EKF, this means storing the parameters in a single state vector and their associated correlations in a covariance matrix. The upshot for EKF-based SLAM is that the vehicle pose estimate and the map feature locations must be stored in the same state vector, and this vector must be augmented as new features are observed and added to the map. Assuming the SLAM process satisfies the basic EKF conditions of near-linearity and approximately Gaussian uncertainty distributions, then the uncertainty model provided by the EKF has been shown to yield monotonic map convergence [42]. In the limit, with repeated observation, the map becomes totally rigid such that the relative feature locations are perfectly known and the global uncertainty approaches a lower bound established by the initial vehicle pose uncertainty.

Stochastic SLAM suffers from three main weaknesses: high computation and storage costs, fragile data association, and inconsistent treatment of non-linearity. High computation and storage are the price paid for maintaining correlations in the state covariance matrix. For n state parameters, the n^2 covariance elements need to be updated with each new observation. Generally, the number of vehicle parameters is insignificant compared to the number of feature parameters, and so the computation and storage are specified as being $O(n^2)$ where n is the number of map features.

Data association failure is a much more serious problem for SLAM than for *a priori* map localisation. Simple localisation may be able to recover from a minor misassociation, because only the vehicle pose estimate is affected, but with SLAM the map is also altered and these inconsistencies tend to be self-propagating, causing divergence. A further data association problem concerns the management of non-associated observations. These are either new map features, outlier measurements, or observations of dynamic objects, and discerning the latter two is essential to prevent cluttering the map. The most difficult data association complication arises during cycle detection. This is difficult because, not only is the vehicle pose uncertain, but the new and old portions of the map are also uncertain in relation to each other, which reduces the reliability even of batch data association.

Non-linearity is typically not a significant problem for normal SLAM operation [48] but, again, proves to be an issue during cycle detection. Incremental map building without cycles is a reasonably linear process and the EKF requirement of small estimation errors is generally met. The same is true for cycles that accrue only small pose errors, as the map feature correlations permit proper error compensation back through the map loop. However, cycles involving large accumulated errors can incur corrections that violate the linearised filter assumptions and result in divergence. There are two aspects that contribute to the inconsistency of large corrections. The first is the effect of linearising a highly non-linear observation model, as the first-order approximation may be inaccurate if the state uncertainty is large compared to the observation uncertainty. This problem could possibly be addressed using iterated linearisation methods such the *iterated EKF* (IEKF) [6, pages 404–406] or the *smoothly constrained Kalman filter* (SCKF) [37]. The second aspect is that the Gaussian uncertainty approximation for the state (i.e., vehicle and feature covariances and correlations) may be optimistic because of the large estimate deviations, which would prevent convergence to the true state.

In summary, feature maps are a viable representation for long-term convergent SLAM in fairly small-scale environments where stable landmarks are observable, computation is tractable, and accumulated state uncertainty does not exceed conservative limits. For feasible and convergent operation in larger areas, modifications to the basic stochastic SLAM algorithm are required.

2.1.3 Topological Maps

Topological maps [85] illustrate a major conceptual shift in environmental representation. Occupancy grids and feature maps are both metric maps where location is defined as a set of coordinates in Cartesian space. Topological maps, however, do not rely on metric measurements and instead represent the environment in terms of places and connecting paths as shown in Figure 2.4. They are generally depicted by a graph structure where

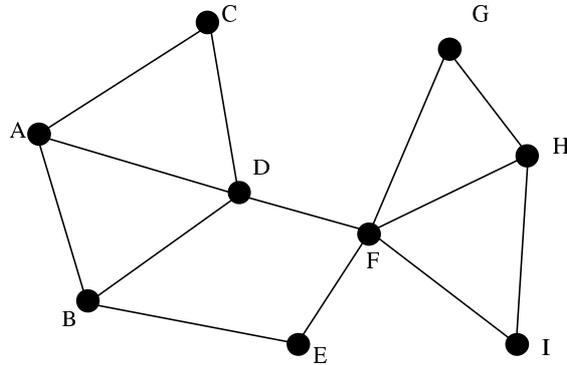


Figure 2.4: Topological map. The environment is defined using a graph data structure where each node (or vertex) contains a place description and each edge contains a path description. Travelling from one place to another entails travelling via intermediate places and standard graph shortest path algorithms can be used (e.g., A to H requires travelling through the sequence A-D-F-H).

the graph nodes define particular locations in the environment (termed *distinctive places*) and the graph edges define procedural information for traveling between nodes. Thus, navigation between two non-adjacent locations is determined by a sequence of transitions between intermediate place nodes. The concept works on the assumptions that distinctive places are locally distinguishable from the surrounding area and the procedural information is sufficient to enable the robot to travel within recognising distance of a specified place.

Place recognition is a form of data association where the observation-to-map correspondence is based on the apparent similarity between two data sets—in this case between the observed information and a graph node description. Appearance-based association, which relies on locally unique data, can be considered in contrast with metric proximity constraints, as used with feature maps, which enables association in the presence of identical landmark data. For place recognition to function correctly, a node description must be unique along the connecting path regions from its adjacent nodes. This allows the robot to compare observed data with the node template while travelling towards it and to determine being “at” the place location once a positive match is made. Most topological localisation systems use one of two basic types of place recognition. The first is based on range-bearing measurements, such as sonar or laser, and involves defining place nodes at locations fitting certain geometric qualities and then matching the measured data to these place descriptions. For example, these descriptions might be observation criteria such as equidistance from near objects [85, 31], or simple models of indoor structures like doors and corridor intersections [83, 3]. The second recognition method is based on vision data and involves defining each place node with images obtained from its (arbitrary) location and then matching observed images to the node according to a prescribed similarity measure [51, 1, 137].

Topological maps, as an *a priori* reference, are attractive for their efficient and compact representation, and their logical organisation for tasks like path planning. The departure

from metric representation makes pose uncertainty estimation irrelevant and, instead, qualitative measures are used like “follow path from A to B” or “at B”. Of particular advantage, is the ability to utilise standard graph algorithms for high-level planning operations such as finding the shortest path between non-adjacent nodes [39].

The primary weakness of topological maps concerns ensuring reliable navigation between places, and subsequent place recognition, without the aid of some form of metric location measure. Travelling between nodes using purely qualitative trajectory information, such as wall following [85], is often sufficient for static structured environments but, in more complex and dynamic environments, may fail to guide the robot to the appropriate place vicinity. The most critical weakness, however, is place recognition. If a place is not recognised (false negative) or an alternate location is mistaken for a place (false positive) then the topological sequence is broken and the robot becomes lost. False negatives occur because of alteration in place appearance through circumstances such as viewpoint variation, occlusion, structural change, dynamic objects or changed lighting conditions. Both geometric and visual recognition methods are sensitive to this form of failure. False positives can be generated if another portion of the environment appears similar to the place definition—meaning that the place is not locally unique. This is common in highly structured environments (such as rows of office cubicles) but can also be a symptom of inadequate place definition. For the simple descriptions offered by most geometric recognition methods, ambiguous associations may be created even by the presence of transient objects. Vision-based recognition is probably more immune to false positives because of the increased level of information defining the node.

One further problem with topological maps is their limitation to waypoint-based navigation. The robot is constrained to follow specific trajectories and to pass through (or very near) each place location. Effectively, control is directly tied to the localisation process (i.e., the robot must perform active localisation). While this situation is adequate for many autonomous vehicle systems, there exist applications where the robot trajectory should be independent of discrete place locations and passive localisation is necessary.

Topological SLAM [85] operates by performing exploration of the environment guided by a set of path following criteria, and recording place descriptions at appropriate locations. For geometric place recognition, these would be locations effecting certain patterns in the sensory data and, for vision-based recognition, they could be either regularly-spaced locations or locations where a given distinctiveness metric is maximised. As each new place is found, it is connected to the previous place according to the path following specifications required to reach it. In this way, the map is built as a *linear* sequence of places, which continues until a place is observed that matches a previously stored place description (presuming the robot is not simply traversing old sections of the map). This matching place description is generated by either a new region of similar appearance or by observing the old place reached via an alternate route. If the match can be identified unequivocally as the old place location, then a cycle² is created, linking the topological sequence back upon itself to form a closed path.

The *a priori* map problems of qualitative path following and sequential data association

²The term *cycle detection*, used in this thesis to refer to map revisitation, is coined from the graph-theoretic term for a closed path. For topological maps, the result of loop closure is a cycle in the map graph.

remain significant for topological SLAM, but the most prominent concern is cycle detection. By avoiding metric location measurement, topological SLAM removes the difficulties of uncertainty representation and non-linearities but, instead, places full responsibility for robust operation upon data association. In the case of cycle detection, where an observed place is found to resemble a previous place (or perhaps several previous places), data association becomes ambiguous, and the observed place could be one of the stored locations or a newly discovered location. Discerning the correct association can be found by a method of *rehearsal* [85], where the ensuing sequence of places is tracked until the number of candidate cycles is reduced to one (a cycle) or none (a new place). This method is appropriate if a globally unique place, or place sequence, exists in the map [9]. However, in many environments, such uniqueness cannot be ascertained, and the cycle cannot be confirmed since the matching sequence could actually represent a similar, but previously unexplored, region of the environment. An approach for artificially creating unique locations by having the robot drop coloured markers has been proposed [46], but this method is not practical for many robot applications. In general, the observation of an expected place sequence serves to increase confidence of a cycle detection but cannot definitely confirm this hypothesis.

Essentially, unless the environment possesses at least one globally unique sequence of places, cycle detection must always be ambiguous. This is the key weakness in the topological map paradigm as environmental similarity may eventually produce a consistently similar sequence of places and result in data association failure. The solution to this problem is to introduce metric information, which would enable the estimation of pose uncertainty between places—bounding the cycle search-space so that place sequences need only be locally unique.

2.1.4 Hybrid Topological-Metric Maps

The qualities of metric and topological maps are complementary. Metric maps, with an appropriate uncertainty representation, constrain data association and permit non-qualitative trajectory planning, while topological maps break the world up into locally connected regions and avoid the problems of maintaining a global reference frame. Hybrid topological-metric maps [141, 121, 30, 31] are basically topological frameworks where the place definitions and/or path definitions contain metric (as well as qualitative) information.³ Importantly, this means that places are no longer restricted to discrete locations but can describe regions of arbitrary size and shape as local metric maps.

Defining places as small-scale feature maps is proposed in this thesis as an effective method for performing large-scale SLAM. The global topological structure addresses the problems of computational cost and non-linearities by decoupling distant (non-adjacent) regions. Cycle detection remains a significant problem but, by combining place recognition methods with pose uncertainty constraints, the ability to disambiguate cycles is greatly improved. Further development of this hybrid map concept is presented in Chapter 6.

³An alternative type of topological-metric hybrid is presented in [128] where the map is essentially a standard (occupancy grid) metric map overlaid with a topological network for path planning. However, this type of map does not address the issues of large-scale localisation with which this thesis is concerned and so is not investigated further.

2.2 The Stochastic SLAM Algorithm

The fundamental equations for feature map SLAM based on the EKF are presented in this section. An account of the EKF algorithm, and a brief introduction to state-space concepts, is provided in Appendix B. For convenience, the k notation used in Appendix B is dropped in the following explanation as the sequence of operations is apparent from its context.

Stochastic SLAM [122, 36, 42, 140] is performed by storing the vehicle pose and map landmarks in a single state vector, and estimating the state parameters via a recursive process of prediction and observation. The prediction stage deals with vehicle motion based on incremental dead reckoning estimates, and increases the uncertainty of the vehicle pose estimate. The observation, or update, stage occurs with the re-observation of stored features, and improves the overall state estimate. When a feature is observed for the first time, however, it is added to the state vector through an initialisation process called state augmentation.

2.2.1 Vehicle, Map and Augmented State Vectors

The vehicle state is represented, in this thesis, by its pose relative to a base Cartesian coordinate frame (as shown in Figure 2.5) with mean and covariance defined as

$$\hat{\mathbf{x}}_v = [\hat{x}_v \quad \hat{y}_v \quad \hat{\phi}_v]^T \quad (2.1)$$

$$\mathbf{P}_v = \begin{bmatrix} \sigma_{x_v x_v}^2 & \sigma_{x_v y_v}^2 & \sigma_{x_v \phi_v}^2 \\ \sigma_{x_v y_v}^2 & \sigma_{y_v y_v}^2 & \sigma_{y_v \phi_v}^2 \\ \sigma_{x_v \phi_v}^2 & \sigma_{y_v \phi_v}^2 & \sigma_{\phi_v \phi_v}^2 \end{bmatrix} \quad (2.2)$$

The locations of 2-D point features observed by the vehicle form a map in the same base coordinate system. (More elaborate parametric feature models, such as lines, might also be used, but are not implemented in this thesis.) The covariance matrix \mathbf{P}_m of this map includes cross-correlation information between the features (i.e., the off-diagonal terms) which captures the dependence of each feature location upon knowledge of the other features in the map. Since the feature locations are static, these correlations will increase with each re-observation as the map becomes increasingly rigid.

$$\hat{\mathbf{x}}_m = [\hat{x}_1 \quad \hat{y}_1 \quad \dots \quad \hat{x}_n \quad \hat{y}_n]^T \quad (2.3)$$

$$\mathbf{P}_m = \begin{bmatrix} \sigma_{x_1 x_1}^2 & \sigma_{x_1 y_1}^2 & \dots & \sigma_{x_1 x_n}^2 & \sigma_{x_1 y_n}^2 \\ \sigma_{x_1 y_1}^2 & \sigma_{y_1 y_1}^2 & \dots & \sigma_{y_1 x_n}^2 & \sigma_{y_1 y_n}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{x_1 x_n}^2 & \sigma_{y_1 x_n}^2 & \dots & \sigma_{x_n x_n}^2 & \sigma_{x_n y_n}^2 \\ \sigma_{x_1 y_n}^2 & \sigma_{y_1 y_n}^2 & \dots & \sigma_{x_n y_n}^2 & \sigma_{y_n y_n}^2 \end{bmatrix} \quad (2.4)$$

The SLAM map is defined by an augmented state vector formed by the concatenation of the vehicle state and the feature map state. This is necessary as consistent SLAM relies

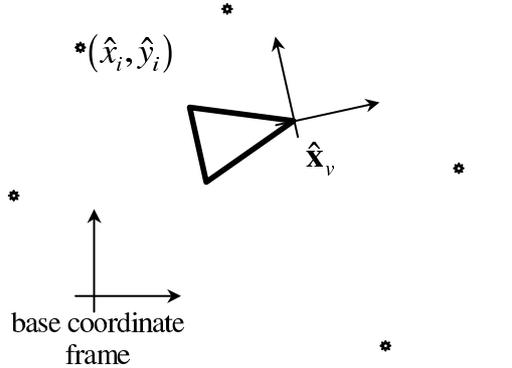


Figure 2.5: Augmented state vector. The SLAM state vector is composed of the vehicle pose and of landmarks observed in the environment referenced with respect to a base coordinate frame.

on the maintenance of correlations \mathbf{P}_{vm} between the vehicle and the map.

$$\hat{\mathbf{x}}_a = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (2.5)$$

$$\mathbf{P}_a = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_m \end{bmatrix} \quad (2.6)$$

Note that the initial condition of the state estimate is usually given as $\hat{\mathbf{x}}_a = \hat{\mathbf{x}}_v = \mathbf{0}$ and $\mathbf{P}_a = \mathbf{P}_v = \mathbf{0}$. In other words, no features have yet been observed and the initial vehicle pose defines the base coordinate origin.

2.2.2 Prediction Stage

The SLAM process model specifies that the vehicle moves relative to its previous pose according to a dead reckoning motion estimate, and the map features remain stationary. The effect of this model on the state estimate is a change in the $\hat{\mathbf{x}}_v$ portion of the state vector, and in the \mathbf{P}_v and \mathbf{P}_{vm} terms of the state covariance matrix, while the $\hat{\mathbf{x}}_m$ and \mathbf{P}_m portions remain constant.

An estimate of the vehicle change-in-pose $\hat{\mathbf{x}}_\delta = [\hat{x}_\delta, \hat{y}_\delta, \hat{\phi}_\delta]^T$ with covariance \mathbf{P}_δ (see Figure 2.6) is commonly obtained using wheel encoder odometry and a vehicle kinematic model. In this thesis, it is obtained using laser-based dead reckoning,⁴ which finds the relative pose between sequential laser scans through a combination of the batch data association algorithm in Chapter 3 and the relative pose estimation method in Appendix C.3.

⁴Concurrent use of laser-based dead reckoning and laser-based SLAM would involve information reuse and lead to an over-optimistic SLAM estimate. However, SLAM may use the laser dead reckoning predict for data association purposes, provided a separate predict step is used as the prior for the SLAM update step. To avoid reuse, the SLAM predict is based on a simple constant velocity dynamic model where the predicted change-in-pose equals the previous observed change-in-pose with suitably large uncertainty to capture vehicle accelerations.

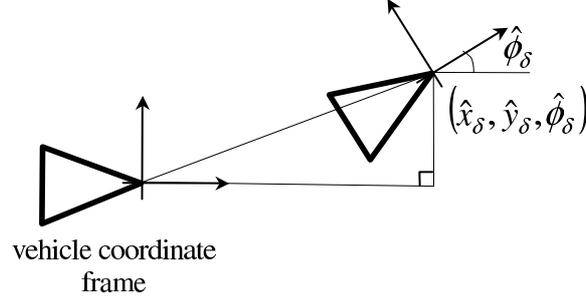


Figure 2.6: Change in pose vector. Laser-based dead reckoning provides a vector of the vehicle motion between sequential scans.

The predicted augmented state, therefore, is given by

$$\hat{\mathbf{x}}_a^- = \mathbf{f}(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_\delta) = \begin{bmatrix} \mathbf{g}(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_\delta) \\ \hat{\mathbf{x}}_m \end{bmatrix} = \begin{bmatrix} \hat{x}_v + \hat{x}_\delta \cos \hat{\phi}_v - \hat{y}_\delta \sin \hat{\phi}_v \\ \hat{y}_v + \hat{x}_\delta \sin \hat{\phi}_v + \hat{y}_\delta \cos \hat{\phi}_v \\ \hat{\phi}_v + \hat{\phi}_\delta \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (2.7)$$

$$\mathbf{P}_a^- = \nabla \mathbf{f}_{\mathbf{x}_a} \mathbf{P}_a \nabla \mathbf{f}_{\mathbf{x}_a}^T + \nabla \mathbf{f}_{\mathbf{x}_\delta} \mathbf{P}_\delta \nabla \mathbf{f}_{\mathbf{x}_\delta}^T \quad (2.8)$$

where the Jacobians $\nabla \mathbf{f}_{\mathbf{x}_a}$ and $\nabla \mathbf{f}_{\mathbf{x}_\delta}$ are defined as

$$\nabla \mathbf{f}_{\mathbf{x}_a} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_a} \right|_{(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_\delta)} = \begin{bmatrix} \nabla \mathbf{g}_{\mathbf{x}_v} & \mathbf{0}_{vm} \\ \mathbf{0}_{vm}^T & \mathbf{I}_m \end{bmatrix}_{(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_\delta)} \quad (2.9)$$

$$\nabla \mathbf{f}_{\mathbf{x}_\delta} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_\delta} \right|_{(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_\delta)} = \begin{bmatrix} \nabla \mathbf{g}_{\mathbf{x}_\delta} \\ \mathbf{0}_{vm}^T \end{bmatrix}_{(\hat{\mathbf{x}}_a, \hat{\mathbf{x}}_\delta)} \quad (2.10)$$

and the Jacobians $\nabla \mathbf{g}_{\mathbf{x}_v}$ and $\nabla \mathbf{g}_{\mathbf{x}_\delta}$ are as follows.

$$\nabla \mathbf{g}_{\mathbf{x}_v} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}_v} \right|_{(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_\delta)} = \begin{bmatrix} 1 & 0 & -\hat{x}_\delta \sin \hat{\phi}_v - \hat{y}_\delta \cos \hat{\phi}_v \\ 0 & 1 & \hat{x}_\delta \cos \hat{\phi}_v - \hat{y}_\delta \sin \hat{\phi}_v \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

$$\nabla \mathbf{g}_{\mathbf{x}_\delta} = \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}_\delta} \right|_{(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_\delta)} = \begin{bmatrix} \cos \hat{\phi}_v & -\sin \hat{\phi}_v & 0 \\ \sin \hat{\phi}_v & \cos \hat{\phi}_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

As these Jacobians only affect the vehicle portion of the covariance matrix \mathbf{P}_v and its cross-correlations \mathbf{P}_{vm} , Equation 2.8 can be implemented more efficiently as

$$\mathbf{P}_a^- = \begin{bmatrix} \nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_v \nabla \mathbf{g}_{\mathbf{x}_v}^T + \nabla \mathbf{g}_{\mathbf{x}_\delta} \mathbf{P}_\delta \nabla \mathbf{g}_{\mathbf{x}_\delta}^T & \nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_{vm} \\ (\nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_{vm})^T & \mathbf{P}_m \end{bmatrix} \quad (2.13)$$

2.2.3 Update Stage

If a feature already stored in the map as estimate (\hat{x}_i, \hat{y}_i) is observed by a range-bearing sensor with the measurement

$$\mathbf{z} = \begin{bmatrix} r \\ \theta \end{bmatrix} \quad (2.14)$$

$$\mathbf{R} = \begin{bmatrix} \sigma_r^2 & \sigma_{r\theta}^2 \\ \sigma_{r\theta}^2 & \sigma_\theta^2 \end{bmatrix} \quad (2.15)$$

where (r, θ) is the range and bearing relative to the observer and \mathbf{R} is the observation covariance, then the sensed information is related to the map by the following equation.

$$\hat{\mathbf{z}}_i = \mathbf{h}_i(\hat{\mathbf{x}}_a) = \begin{bmatrix} \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_i - \hat{y}_v}{\hat{x}_i - \hat{x}_v}\right) - \hat{\phi}_v \end{bmatrix} \quad (2.16)$$

Assuming correct data association of the observation \mathbf{z} to the map feature estimate (\hat{x}_i, \hat{y}_i) , the Kalman gain \mathbf{W}_i can be determined as

$$\nu_i = \mathbf{z} - \mathbf{h}_i(\hat{\mathbf{x}}_a^-) \quad (2.17)$$

$$\mathbf{S}_i = \nabla \mathbf{h}_{\mathbf{x}_a} \mathbf{P}_a^- \nabla \mathbf{h}_{\mathbf{x}_a}^T + \mathbf{R} \quad (2.18)$$

$$\mathbf{W}_i = \mathbf{P}_a^- \nabla \mathbf{h}_{\mathbf{x}_a}^T \mathbf{S}_i^{-1} \quad (2.19)$$

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_a}$ is given by

$$\nabla \mathbf{h}_{\mathbf{x}_a} = \left. \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_a} \right|_{\hat{\mathbf{x}}_a^-} = \begin{bmatrix} -\frac{\Delta x}{d} & -\frac{\Delta y}{d} & 0 & 0 & \dots & 0 & \frac{\Delta x}{d} & \frac{\Delta y}{d} & 0 & \dots & 0 \\ \frac{\Delta y}{d^2} & -\frac{\Delta x}{d^2} & -1 & 0 & \dots & 0 & -\frac{\Delta y}{d^2} & \frac{\Delta x}{d^2} & 0 & \dots & 0 \end{bmatrix} \quad (2.20)$$

$$\Delta x = \hat{x}_i - \hat{x}_v$$

$$\Delta y = \hat{y}_i - \hat{y}_v$$

$$d = \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2}$$

For SLAM maps with large numbers of features, the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_a}$ consists mainly of zero terms enabling efficient calculations of Equations 2.18 and 2.19. The non-zero terms align with the positions of the vehicle states and the observed feature states (\hat{x}_i, \hat{y}_i) in the augmented state vector. The *a posteriori* SLAM estimate is subsequently determined from the update equations.

$$\hat{\mathbf{x}}_a^+ = \hat{\mathbf{x}}_a^- + \mathbf{W}_i \nu_i \quad (2.21)$$

$$\mathbf{P}_a^+ = \mathbf{P}_a^- - \mathbf{W}_i \mathbf{S}_i \mathbf{W}_i^T \quad (2.22)$$

The observation model in Equation 2.16 correlates the feature estimate to the vehicle pose estimate and serves to reduce the uncertainty of both. Through correlation to the vehicle pose estimate, the map features become correlated to each other and these correlations increase monotonically until their locations (relative to each other) become perfectly known.

On a practical note, if several independent observations are available at once (a batch observation $\mathbf{z} = [r_1, \theta_1, \dots, r_n, \theta_n]^T$), then a more accurate update may be possible if they are treated corporately than if each observation is processed individually. The reason for improved performance is because the EKF performs linearised error correction and the update tends to be “pulled in the right direction” better if the innovation vector ν consists of several observation errors at once. The difference in performance is most noticeable if the vehicle pose is very uncertain prior to the update stage. However, a disadvantage of batch update processing is the need to invert the innovation covariance matrix \mathbf{S} which requires computation to the order $O(n^3)$ where n is the number of features in the batch. A simple compromise is to process the observations in manageable sub-batches.

2.2.4 State Augmentation

As the environment is explored, new features are observed and must be added to the stored map. A method for initialising new features is shown below.⁵ First, the state vector and covariance matrix are extended (augmented) with the polar values of the new observation \mathbf{z} , and its covariance \mathbf{R} , as measured relative to the observer.

$$\hat{\mathbf{x}}_{aug} = \begin{bmatrix} \hat{\mathbf{x}}_a \\ \mathbf{z} \end{bmatrix} \quad (2.23)$$

$$\mathbf{P}_{aug} = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{vm} & \mathbf{0} \\ \mathbf{P}_{vm}^T & \mathbf{P}_m & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix} \quad (2.24)$$

A function \mathbf{g}_i is derived to convert the polar observation \mathbf{z} to a global Cartesian feature location. This transformation is a function of the new observation and the current vehicle pose.

$$\mathbf{g}_i(\mathbf{x}_v, \mathbf{z}) = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_v + r \cos(\theta + \phi_v) \\ y_v + r \sin(\theta + \phi_v) \end{bmatrix} \quad (2.25)$$

The augmented state can then be initialised to the correct values by performing a linearised transformation by the function \mathbf{f}_i as follows.

$$\hat{\mathbf{x}}_a^+ = \mathbf{f}_i(\hat{\mathbf{x}}_{aug}) = \begin{bmatrix} \hat{\mathbf{x}}_a \\ \mathbf{g}_i(\hat{\mathbf{x}}_v, \mathbf{z}) \end{bmatrix} \quad (2.26)$$

$$\mathbf{P}_a^+ = \nabla \mathbf{f}_{\mathbf{x}_{aug}} \mathbf{P}_{aug} \nabla \mathbf{f}_{\mathbf{x}_{aug}}^T \quad (2.27)$$

where the Jacobian $\nabla \mathbf{f}_{\mathbf{x}_{aug}}$ is given by

$$\nabla \mathbf{f}_{\mathbf{x}_{aug}} = \left. \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_{aug}} \right|_{\hat{\mathbf{x}}_{aug}} = \begin{bmatrix} \mathbf{I}_v & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m & \mathbf{0} \\ \nabla \mathbf{g}_{\mathbf{x}_v} & \mathbf{0} & \nabla \mathbf{g}_{\mathbf{z}} \end{bmatrix} \quad (2.28)$$

⁵Thanks to Stefan Williams for this derivation.

and the Jacobians $\nabla \mathbf{g}_{\mathbf{x}_v}$ and $\nabla \mathbf{g}_{\mathbf{z}}$ are as follows.

$$\nabla \mathbf{g}_{\mathbf{x}_v} = \left. \frac{\partial \mathbf{g}_i}{\partial \mathbf{x}_v} \right|_{(\hat{\mathbf{x}}_v, \mathbf{z})} = \begin{bmatrix} 1 & 0 & -r \sin(\theta + \hat{\phi}_v) \\ 0 & 1 & r \cos(\theta + \hat{\phi}_v) \end{bmatrix} \quad (2.29)$$

$$\nabla \mathbf{g}_{\mathbf{z}} = \left. \frac{\partial \mathbf{g}_i}{\partial \mathbf{z}} \right|_{(\hat{\mathbf{x}}_v, \mathbf{z})} = \begin{bmatrix} \cos(\theta + \hat{\phi}_v) & -r \sin(\theta + \hat{\phi}_v) \\ \sin(\theta + \hat{\phi}_v) & r \cos(\theta + \hat{\phi}_v) \end{bmatrix} \quad (2.30)$$

The matrix multiplication in Equation 2.27 requires computation to the order $O(n^3)$ where n is the number of features in the map. Due to the sparseness of the Jacobian matrix, however, a much more efficient implementation is possible as the transform only affects the block diagonal matrix of the new feature and its off diagonal cross-correlations to the rest of the map.

$$\mathbf{P}_a^+ = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{vm} & \mathbf{P}_v \nabla \mathbf{g}_{\mathbf{x}_v}^T \\ \mathbf{P}_{vm}^T & \mathbf{P}_m & \mathbf{P}_{vm}^T \nabla \mathbf{g}_{\mathbf{x}_v}^T \\ \nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_v & \nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_{vm} & \nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_v \nabla \mathbf{g}_{\mathbf{x}_v}^T + \nabla \mathbf{g}_{\mathbf{z}} \mathbf{R} \nabla \mathbf{g}_{\mathbf{z}}^T \end{bmatrix} \quad (2.31)$$

Feature deletion from the SLAM map is straightforward. The elements are simply removed from the state vector and the associated rows and columns are deleted from the state covariance matrix. Map management through the deletion of features is discussed at length in Chapter 5.

2.2.5 Efficiency Improvements for Large-Scale SLAM

The state covariance update in Equation 2.22 is an $O(n^2)$ computation for n map features. For tractable operation in large-scale environments, several efficient variations of the stochastic SLAM algorithm have been proposed.

The *compressed filter* [65] limits the SLAM update stage to only affect features within a bounded region surrounding the vehicle location. As such, the computational cost becomes $O(n_a^2)$ where n_a is the number of features contained within the local region. Correlations to features outside the local region are preserved in an optimal manner but the $O(n^2)$ transfer of this information to the entire map is only performed when the vehicle transitions to another region. Thus a full $O(n^2)$ SLAM update can be processed as a low frequency event even with high frequency observation information. Moreover, the full update can be handled as a background task with only the features within the new region requiring immediate update in sequence with incoming observations. A further contribution in [65] is a suboptimal algorithm for the full update step that permits an upper bound to be set on the number of features receiving a decrease in uncertainty—therefore bounding the quadratic portion of the SLAM update. With appropriate map organisation, this method is shown to produce results almost indistinguishable from the optimal algorithm.

The *postponement* algorithm [81] is similar to the compressed filter. It is an optimal filter, and permits the same $O(n_a^2)$ local region complexity and deferred application of the full $O(n^2)$ update while the vehicle remains within the local area. Unlike the compressed filter, the postponement local regions are not governed by geometric bounds but can expand dynamically to accommodate newly observed features and extend the region bounds.

Thus, a local region is described as a “*cache* of features whose update is cheap, but which ultimately must be ‘swapped out’.” A full update is required only when the computational load of the local region becomes excessive, whereupon a new local region is commenced and the full update may be performed as a background task.

The *constrained local submap filter* (CLSF) [140] is an alternative approach which performs actual information gathering and localisation via the construction of an independent local map. This technique operates as follows. The robot is at some (approximately) known location in the global map and, from this location, starts a new independent SLAM map. As the robot moves and receives observations of the environment the new local map is built in the normal SLAM fashion and the global map serves as a static map. Thus, the robot pose in global coordinates may be found by transforming the local pose estimate given the known global pose of the local map coordinate origin. When the local map reaches a state where either it contains too many features or the vehicle location is too uncertain, the local map is registered with the global map and a new local map is commenced. The process of local map registration results in a global map update equivalent to the standard optimal SLAM algorithm. Like the compressed filter, this method enables a low frequency global SLAM update that can be performed as a background task (i.e., the full update does not need to reach completion until the next local map is ready for registration). An additional benefit of this approach is that it can be applied directly to multiple robot systems where each robot builds its own local map and periodically registers it with a shared global map.

Another recent technique called *local map sequencing* [127] performs local map construction and registration in a manner virtually identical to the CLSF. The robot constructs local submaps, and these are periodically registered with the global map by transforming to the global coordinate frame and augmenting the global state vector. Common features between the local and global maps are fused via geometric constraints and duplicate features are subsequently removed to produce a single map equivalent to the full SLAM solution.

All of the above proposals improve the computational burden of stochastic SLAM but they do not address the $O(n^2)$ memory requirement, and storage may eventually present a problem. The *covariance intersection* [77] is an interesting development of the (extended) Kalman filter that enables the consistent fusion of information without knowing the correlation between state variables or between sources of observed data. Effectively, the covariance intersection filter performs data fusion by producing the best possible estimate given an assumption of worst case conditions where all available information is fully correlated. This filter is applied to the SLAM problem in [135] and results in a formulation where both computation and storage are $O(n)$. However, the assumption of total state and observation dependence is highly suboptimal and the SLAM map tends to suffer large feature uncertainty within a short distance from the map origin. The map also tends to converge only very slowly or not at all with repeated observation.

2.3 Experiments in Outdoor SLAM

One of the main contributions of this thesis is verification of the techniques developed in the later chapters through experimental application in a variety of rugged outdoor environments. These experiments reveal several factors influencing outdoor SLAM, in addition to scale-related issues, which make it considerably more difficult than indoor SLAM.



Figure 2.7: Two scanning lasers (different models) attached to the front bumper of a standard utility vehicle.

Most indoor environments permit reasonably straightforward SLAM implementation because they possess a smooth planar ground surface and a high level of man-made structure. Smooth ground surface enables the use of accurate wheel-encoder odometry and the surface flatness means that the system closely resembles the assumed 2-D world model. Sensor observations of man-made structure, such as walls, are often well modelled by geometric primitives, such as lines and corners, and produce reliable static features. A further simplification for indoor SLAM is the relative low speed of indoor vehicles, which means that the vehicle motion is kinematic rather than dynamic, the sensor scan does not suffer appreciable distortion, and the effects of timestamp discrepancies are negligible.

Outdoor environments are much less ideal and this section discusses the practical issues concerning outdoor SLAM in non-flat non-smooth environments. The particular context of this discussion is derived from experiments using a high-speed road vehicle equipped with a scanning range laser sensor. Finally, some comments are made regarding the role of GPS in outdoor localisation.

2.3.1 SLAM with a Scanning Range Laser

The experimental data used in this thesis was obtained from a 2-D scanning range laser (SICK PLS) mounted on the front bumper of a standard motor vehicle (see Figure 2.7). The laser returns a 180° planar sweep of range measurements in 0.5° intervals (i.e., 361 range values in anti-clockwise order) with a range resolution of about $\pm 50mm$. The vehicle was also equipped with wheel and steering encoders, but this information was used for comparison purposes only.

Scanning laser was the sole information source for all experimentation in this thesis (SLAM and dead reckoning). This meant that successful operation required the existence

of visible static objects in the laser field-of-view, and these tests would certainly fail in environments that do not meet this basic criterion. It is important to realise that these experiments do not seek to present a complete and infallible SLAM system but to demonstrate a single application of the theoretical methods developed in this thesis. System integrity, on the other hand, is dependent on multiple independent and redundant sub-systems with different physical sensing properties and statistically observable failure conditions [117].

A planar scanning laser is a rather inappropriate sensor for outdoor environments, with the main impediment to adequate interpretation of the laser data being non-planar sensor movement—laser tilt caused by acceleration of the vehicle and undulations in the ground surface. This effectively alters the laser view plane so that it observes varying cross-sections of the environment. Thus, objects can appear and disappear erratically from the sensor field-of-view, and persisting objects can appear to change shape or shift location. A particularly troublesome problem is *ground sweeps*, where the laser dips towards the ground and the viewed scene becomes intermittently swamped with ground returns.

2.3.2 Two Dimensional Projection of Non-Planar Environments

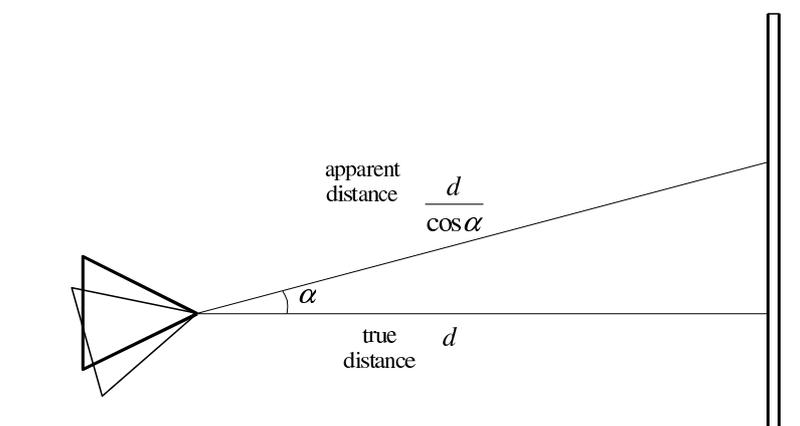
In this thesis, the world is modelled as planar and the vehicle pose is defined by position and heading (x, y, ϕ) . The three-dimensional components of non-flat ground surfaces, therefore, result a distorted representation of the environment. For objects observed by a scanning laser sensor, the effect of 3-D motion is most pronounced for angular tilt as shown in Figure 2.8. In these diagrams, the sensed objects are vertical cylinders and the laser is rotated about the y-axis only, resulting in increasing range measurement distortion with increasing distance in the direction of the x-axis. These biased measurements not only distort the SLAM map but also hinder data association, particularly for more distant objects. Most significantly, these biases are correlated for all feature measurements in a scan, which invalidates the assumption of independent observation errors.

To compensate for tilt distortion, an explicit estimate of sensor tilt may be required to project the essentially 3-D measurements onto the 2-D map plane. However, if the observed features are non-vertical or have variable cross-sections, this correction would provide little benefit and a more complete 3-D model may become necessary.

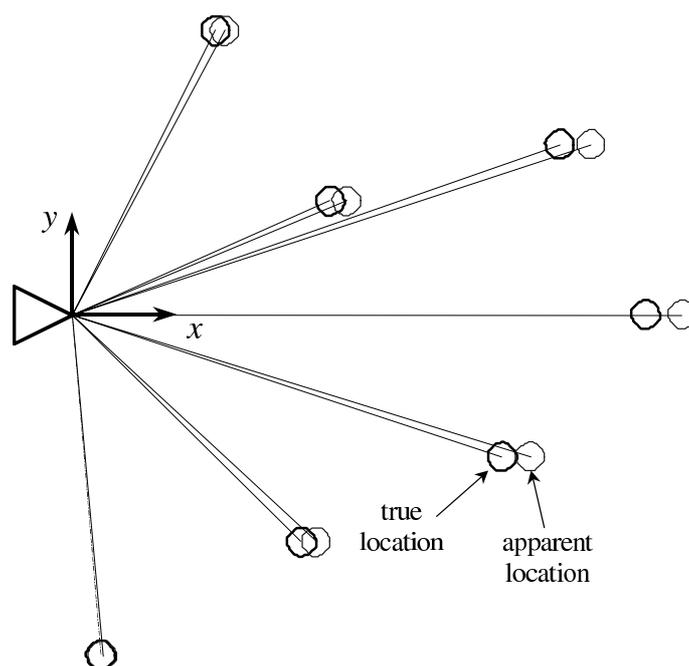
In this thesis, 3-D distortions, and other unmodelled factors such as imprecise feature classification, are incorporated into the observation model by expanding the observation uncertainty to cater for expected worst-case environment conditions. This means that the actual observation uncertainties used in practice are much greater than would be presumed from the sensor accuracy alone. Also, in regions that more closely approximate the 2-D ideal, the actual pose error tends to be considerably less than the estimated uncertainty.

2.3.3 Dead Reckoning in Rough Terrain

Wheel-encoder based odometry becomes unreliable in outdoor terrain due to the combined influence of rugged ground surface and vehicle dynamics, which can invalidate the kinematic model assumptions of pure rolling motion (i.e., that all motion is due to wheel rotation). Wheel slip through loss of traction, skidding, or sliding results in a biased motion estimate and, at high speeds or over rough terrain, the amount of slip can be significant, particularly



(a) Side view



(b) Plan view

Figure 2.8: Trigonometric distortion. The side view (a) shows the effect of tilting the sensor when viewing a vertical object—the apparent distance to the object increases. In plan view (b), the objects appear to recede in the direction of the view-plane incline (the x -axis in this case) causing distorted geometry.

for heavier (high inertia) vehicles. For vehicles with high-slip steering mechanisms, such as articulated skid steering or tracked differential steering, slip can be a problem even at low speeds. Fundamentally, reasonable modelling of wheel slip error is difficult given the complex physics of ground-wheel interaction.

The shape of the terrain is another biasing factor for odometry since, when modelling the world as a plane, a non-flat ground surface causes the apparent distance travelled to be greater than the actual planar motion.

Augmenting odometry with an *inertial navigation system* (INS) can significantly alleviate these problems by detecting slip [116, 86], tilt, and non-planar terrain [56]. Also, odometry and INS are complementary since encoder-based constraint information facilitates on-line attitude alignment of the INS platform [41]. However, while a low-cost INS is able to improve certain high-frequency odometry faults, it cannot observe low-frequency faults such as model biases. Thus, INS-encoder dead reckoning is suited to short-term estimates between external (map-based) observations, but is not sufficient for accurate dead reckoning in the longer term.

An alternative to odometry is dead reckoning based on the incremental pose changes obtained from external sensors (e.g., optical flow [11]). Sensor-based dead reckoning removes dependence on precise kinematic modelling and, compared to odometry, the estimate errors tend to be less prone to bias. This thesis presents a laser-based dead reckoning method which estimates the relative pose between sequential laser scans using the batch data association algorithm in Chapter 3. From scan to scan, the predicted relative pose is determined by a simple dynamic (constant velocity) model, which states that, for fixed time intervals, the expected change-in-pose equals the previous change-in-pose. Thus, the data association search-space is constrained by rough maximum acceleration bounds about the predicted value and a dead reckoning estimate can be computed provided the sequential scans possess a subset of common landmark information.

Sensor-based and odometric dead reckoning are not mutually exclusive and, used together, can provide a level of redundancy. That is, agreeing estimates from the two systems can be fused while disparities can signal faults such as wheel slip or feature tracking failure.

2.3.4 Ramifications of GPS

GPS (Global Positioning System) [126, 62, 6] is a satellite based localisation system that is capable of providing a three-dimensional global location estimate at almost any point on (or above) the planet surface. It has important implications for outdoor SLAM because, at face value, it would appear to make this line of research irrelevant.

As a localisation mechanism, GPS is essentially an artificial (active) beacon *a priori* map. Observation of the satellites enables position estimation via triangulation, with accuracy of about 10 metres influenced by factors such as atmospheric conditions and geometry of the satellite configuration. The estimate errors are also temporally correlated, consisting of a noise component and a drifting bias component. A dramatic reduction in error is possible through the use of a differential base station (a nearby stationary GPS receiver that transmits GPS error). Differential GPS (DGPS) compensates for the noise and bias components of the GPS measurement enabling improved precision; standard DGPS is accurate to 1–2 metres and real-time kinematic DGPS to 1–2 centimetres.

Exclusive application of GPS for all outdoor localisation purposes is hindered, however, by the following problems. First, reception of the satellite signals requires direct line-of-sight to the GPS receiver. This prevents GPS-based localisation for underground or subsea operations. It also means that GPS can often provide only intermittent coverage in areas containing tall structures like buildings or trees. Second, the level of precision necessary for autonomous navigation (less than one metre) requires simultaneous line-of-sight of at least five satellites. This is again hampered by the presence of occluding structures. Third, satellite signal reflection off objects in the environment can result in *multipath* errors, where the estimated satellite distance is greater than its true distance.⁶ Fault detection alone is reason enough for alternative redundant localisation methods. Finally, since GPS is not self-contained within the mobile robot system, localisation is dependent on uncontrollable external factors such as satellite availability⁷ and possible system downtime.

In environments where GPS is at least intermittently available, it remains a very useful localisation tool since the position estimates it provides are known to have bounded error. These estimates might be used in conjunction with other localisation or SLAM systems to implement smooth and continuous navigation on a very large scale (e.g., see Section 5.2). However, in environments where GPS is unavailable or insufficient (as an *a priori* map), or where multiple redundant navigation loops are required, either alternative maps must be built or the robot must build its own using SLAM.

2.4 Summary

This chapter introduces the fundamental background information for performing SLAM. It presents the four most common map representations: occupancy grids, feature maps, topological maps, and hybrid topological-metric maps; and discusses their relative merits and utility for SLAM. In particular, it contends that occupancy grids are inappropriate for long-term SLAM (the map will drift), feature maps are suited to small-scale SLAM, and a topological-metric hybrid is the most likely answer for SLAM in larger environments.

The traditional algorithm for stochastic SLAM is presented based on feature maps and the EKF. This algorithm requires $O(n^2)$ computation and storage for a map of n features, and so several techniques are reviewed for improving its efficiency.

The experimental context of this thesis is described, where a standard road vehicle is fitted with a scanning range laser and driven through a series of outdoor environments. The issues relating to outdoor SLAM with a laser sensor involve high speeds, uneven ground surfaces, sensor tilt, geometric distortion, and degraded odometry. Finally, given the common availability of GPS, it is argued that research into large-scale SLAM is still relevant, since GPS may not always operate (reliably) in all areas, and some applications require multiple redundant navigation loops.

⁶Some of these problems may not remain in the long term, as scheduled changes to GPS over the next few decades are expected to improve position accuracy and reduce susceptibility to signal blockage and multipath [62, pages 71–76].

⁷Satellite availability refers to the availability of a sufficient number of satellite signals within the receiver line-of-sight. It does not refer to “selective availability”—the intentional introduction of errors into the GPS signals for civilian use. For one, selective availability errors are implicitly removed when using DGPS. Furthermore, selective availability has “been discontinued since May 1, 2000. The prevention of the hostile use of GPS is [now] accomplished through other measures, such as selective and local denial of GPS signals” [6].

Chapter 3

Batch Data Association for Correlated Feature Sets

Data association is arguably the most critical aspect of the SLAM algorithm. Correct correspondence of sensed feature observations to map landmarks is essential for consistent map construction, and a single false match may invalidate the entire process.

Most feature-based SLAM implementations model landmarks as simple geometric primitives, such as points or lines, which means that the maps are represented by sets of virtually identical features, distinguishable only by their locations. This representation lends itself to the data association methods developed for target tracking where correspondence is constrained by statistical geometric information. However, general target tracking problems presume that the targets move independently and the observation-to-target associations are processed without considering possible correlations in target motions. The implication for SLAM is that each observation is processed individually and association is based on uncertainties in the measurement, landmark locations, and vehicle pose. Importantly, if the vehicle pose is highly uncertain (compared to the density of targets) then data association becomes very fragile indeed.

This chapter investigates the target tracking data association problem where the target motions have known correlation. Specifically, the SLAM problem states that the landmarks have zero relative motion and, therefore, become increasingly correlated with repeated observations. Given a set of observations, the constraint information available when processed as a batch can greatly increase data association robustness. This chapter presents the following topics regarding the feature-based data association problem.

- Current target tracking data association methods, for single and multiple targets, are reviewed. In particular, the traditional methods for managing ambiguous associations are discussed.
- The technique of batch data association for correlated targets is introduced as a mechanism for reducing association ambiguity.
- The *combined constraint data association* (CCDA) algorithm is presented. This graph theoretic method constrains association with all available information to produce

mutually compatible association sets. Reliable data association becomes possible even with total uncertainty in observer pose.

- The CCDA algorithm is applied to real laser data to permit sensor-based dead reckoning in outdoor environments.

3.1 Data Association For Target Tracking

Data association when tracking identical targets is determined by the correspondence between the observation measurement and the predicted observation for a given target. This section discusses the statistical measure most commonly used for gauging the valid neighbourhood for association, and states when this constraint is not sufficient for discerning unique associations. For ambiguous associations, there exist a variety of track management methods in the target tracking literature [14, 6]. These methods have been developed from tracking single targets, amidst clutter of spurious measurements, to tracking multiple targets with batch observations in clutter. It should be remembered that none of the methods presented in this section utilise the correlation information that may be available between the targets.

3.1.1 Validation Gating

Since, within the EKF framework, both the target location estimates and target observations are assumed to have Gaussian uncertainty (with infinite tails), it is possible that any given observation might correspond to any target. However, to reject unlikely associations, only targets within a reasonable neighbourhood of an observation should be considered. Association validation is usually performed in observation space, and a *validation gate* defines the maximum permissible discrepancy between a measurement \mathbf{z}_i and a predicted observation $\hat{\mathbf{z}}_j = \mathbf{h}(\hat{\mathbf{x}}_j)$ for target \mathbf{x}_j .

The most common statistical validation gate is based on the *normalised innovation squared* (NIS) [6] also known as the *Mahalanobis distance* [45]. Given an observation innovation $\nu_{ij} = \mathbf{z}_i - \mathbf{h}(\hat{\mathbf{x}}_j)$ with covariance \mathbf{S}_{ij} , the NIS is defined as

$$M_{ij} = \nu_{ij}^T \mathbf{S}_{ij}^{-1} \nu_{ij} \quad (3.1)$$

For a Gaussian distributed innovation sequence, the NIS forms a χ^2 (chi-squared) distribution. The shape of the χ^2 distribution is dependent on the dimension of the innovation vector as shown in Figure 3.1. The gate, therefore, is applied as a maximum NIS threshold $M_{ij} < \gamma_n$, where the innovation is of dimension n . The integral of the χ^2 distribution from 0 to γ_n specifies the probability that, if \mathbf{z}_i is truly an observation of target \mathbf{x}_j , the association will be accepted.¹

The NIS validation gate remains the underlying association threshold for the batch data association method presented in this chapter.

¹It is worth noting that the NIS gate does not actually say anything about the rejection of false associations; it only specifies the probability of accepting (or rejecting) correct associations. Thus, although used as a rejection mechanism, it does not explicitly define a statistical measure of false association rejection.

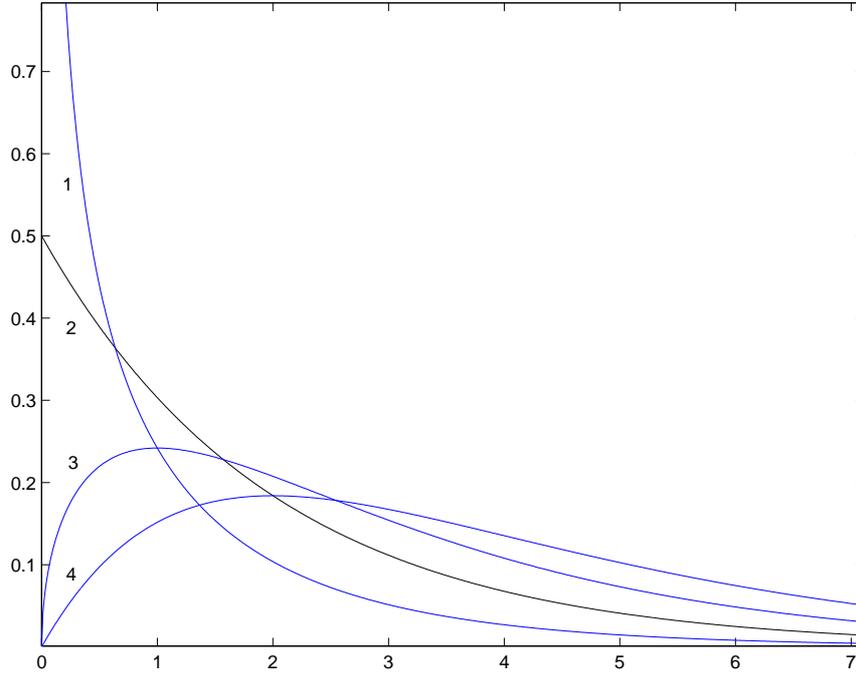


Figure 3.1: χ^2 distribution. The shape of this distribution, shown here for 1 to 4 degrees of freedom, is determined by the dimension of the innovation vector. The integral of this distribution from 0 to γ_n defines the probability of accepting a correct association.

Example 3.1

Associating a range-bearing measurement $\mathbf{z} = [r, \theta]^T$ to a point target $\mathbf{x} = [x, y]^T$. The predicted observation is found from the observation model

$$\mathbf{z} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \arctan\left(\frac{y}{x}\right) \end{bmatrix}$$

If the target estimate is $\hat{\mathbf{x}}$ with covariance \mathbf{P} and the observation covariance is modelled as \mathbf{R} , then the observation innovation $\nu = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}})$ has covariance

$$\mathbf{S} = \nabla \mathbf{h}_{\mathbf{x}} \mathbf{P} \nabla \mathbf{h}_{\mathbf{x}}^T + \mathbf{R}$$

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}}$ is

$$\nabla \mathbf{h}_{\mathbf{x}} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\hat{x}}{\sqrt{\hat{x}^2 + \hat{y}^2}} & \frac{\hat{y}}{\sqrt{\hat{x}^2 + \hat{y}^2}} \\ -\frac{\hat{y}}{\hat{x}^2 + \hat{y}^2} & \frac{\hat{x}}{\hat{x}^2 + \hat{y}^2} \end{bmatrix}$$

With the innovation vector being of dimension 2, the gate

$$\nu^T \mathbf{S}^{-1} \nu < \gamma_2 = 6.0$$

will accept 95% of correct associations, as 95% of the χ^2 probability mass lies between 0 and 6.0.

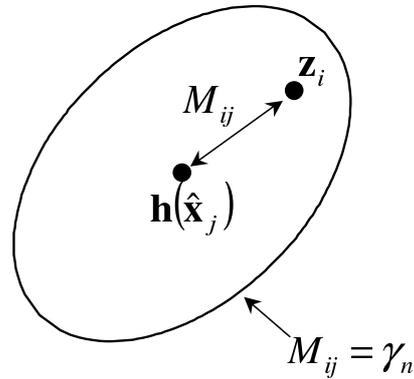


Figure 3.2: Validation gate. The NIS of the measurement \mathbf{z}_i and the predicted observation $\mathbf{h}(\hat{\mathbf{x}}_j)$ must be less than γ_n to be considered as a candidate assignment.

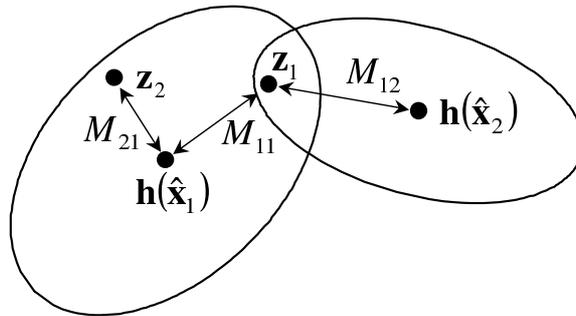


Figure 3.3: Ambiguous data association. If either a single observation falls within the validation gate of multiple targets, or multiple observations fall within the validation gate of a single target, there exists uncertainty as to the correct assignments. For example, either \mathbf{z}_1 or \mathbf{z}_2 could be assigned to $\hat{\mathbf{x}}_1$, and \mathbf{z}_1 could be assigned to either $\hat{\mathbf{x}}_1$ or $\hat{\mathbf{x}}_2$.

The validation gate may be visualised as an ellipsoid in observation space centred about the predicted observation $\mathbf{h}(\hat{\mathbf{x}}_j)$ as shown in Figure 3.2. An acceptable observation must fall within this ellipse. Data association ambiguity occurs if either multiple observations fall within the validation gates of a particular target, or a single observation lies within the gates of multiple targets (see Figure 3.3). Furthermore, it is possible that an observation might arise from clutter or non-tracked targets leading to false associations even with the satisfaction of unique gating conditions.

The simplest method for reducing ambiguous associations is to reduce the validation gate threshold γ_n . This increases the rejection ratio of outlier measurements but, effectively, serves to ignore good information while still allowing false associations to slip through. In recent target tracking literature, a number of mechanisms have been developed to appropriately deal with data association ambiguity, while maintaining suitable gate thresholds (95% acceptance of true measurements, for example).

3.1.2 Tracking a Single Target in Clutter

The most basic form of target tracking is estimating the motion of a single point target with the presence of spurious measurements or clutter.² Single target tracking introduces several ambiguity management methods that are later developed for multiple target, multiple observation problems.

The most common ambiguity resolution method is *nearest neighbour* data association. Given a set of observations $\mathbf{z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ within the validation gate of target \mathbf{x} , a likelihood of association Λ_i can be calculated for each $\mathbf{z}_i \in \mathbf{z}$.

$$\Lambda_i = \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{S}_i|}} \exp\left(-\frac{1}{2} \nu_i^T \mathbf{S}_i^{-1} \nu_i\right) \quad (3.2)$$

where n is the dimension of the innovation vector. This is simply a likelihood function of Gaussian contact [125, page 35]. An equivalent metric, the normalised distance N_i , is obtained by taking logs of Equation 3.2.

$$N_i = \nu_i^T \mathbf{S}_i^{-1} \nu_i + \ln |\mathbf{S}_i| \quad (3.3)$$

Nearest neighbour data association then selects the observation that minimises N_i (or maximises Λ_i). Note that nearest neighbour is sometimes referred to as *maximum likelihood* data association because of the equivalence of Equations 3.2 and 3.3. In environments where the density of clutter is much less than the target location and measurement uncertainties, nearest neighbour works quite well. However, its performance deteriorates rapidly in environments with relatively high clutter density. In practical terms, nearest neighbour offers similar performance to reduction of the validation gate threshold.

A method designed to explicitly incorporate the informational uncertainty induced by ambiguous associations is *probabilistic data association* (PDA) [7]. Also known as the *all-neighbours* approach, PDA combines the information from all the candidate associations as a weighted average of the individual updates. The uncertainty of this combined update is modified to represent the association uncertainty.

A third ambiguity management approach is *multiple hypothesis tracking* (MHT) [113]. MHT delays the application of hard observation-to-target assignment by forming an individual track for each ambiguous association. This method, termed *track splitting*, views each candidate observation as a viable track hypothesis (plus a further hypothesis that all observations in the gate are false). With subsequent observations, new hypotheses are formed and old ones are pruned based on a likelihood function. The effect of MHT is to defer irrevocable data association decisions until further information arises that removes the ambiguity (i.e., through compatibility with subsequent motion observations).

3.1.3 Tracking Multiple Targets in Clutter

Multiple target tracking with multiple simultaneous observations develops the target tracking problem to deal with ambiguities due to both spurious measurements and known targets.

²Clutter is defined as the presence of false observation returns. Typically these are assumed to form a Poisson distribution within the validation gate of the predicted target location. If, however, the returns are not random occurrences but are produced by *discrete interfering sources* [55], then they do not behave as clutter and should be tracked as targets in their own right.

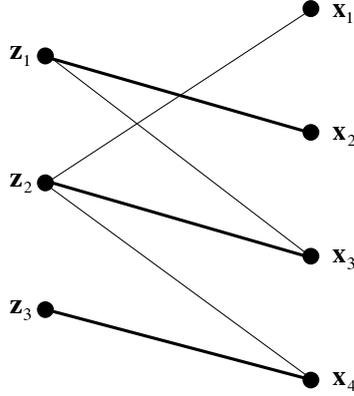


Figure 3.4: Injective assignment example. The validated observation-to-target associations are indicated by connecting lines. The set $\{E_1, \dots, E_5\}$ of all maximal assignment sets complying with the injective mapping constraint is as follows (E_1 is pictured in bold above):
 $E_1 = \{(\mathbf{z}_1, \mathbf{x}_2), (\mathbf{z}_2, \mathbf{x}_3), (\mathbf{z}_3, \mathbf{x}_4)\}$, $E_2 = \{(\mathbf{z}_1, \mathbf{x}_2), (\mathbf{z}_2, \mathbf{x}_1), (\mathbf{z}_3, \mathbf{x}_4)\}$,
 $E_3 = \{(\mathbf{z}_1, \mathbf{x}_3), (\mathbf{z}_2, \mathbf{x}_1), (\mathbf{z}_3, \mathbf{x}_4)\}$, $E_4 = \{(\mathbf{z}_1, \mathbf{x}_2), (\mathbf{z}_2, \mathbf{x}_4)\}$, $E_5 = \{(\mathbf{z}_1, \mathbf{x}_3), (\mathbf{z}_2, \mathbf{x}_4)\}$.

The main extension introduced with multiple target tracking is enforcement of the injective (or one-to-one) mapping constraint, such that no two observations in a batch may be assigned to the same target. In other words, each observation must map to a unique target or be classified as unknown (clutter or new target).

The multiple target equivalent of nearest neighbour data association is *optimal assignment*, which operates as follows. Let $E_k = \{e_1, \dots, e_n\}$ represent a maximal set of compatible (injective) assignment pairs such that $e_m = \{i, j\}$ stands for assignment \mathbf{z}_i to \mathbf{x}_j (see Figure 3.4, for example). Notice that not all E_k are the same size but are maximal in the sense that, given the assignments already contained in the set, no further assignments can be added. Optimal assignment finds the set E_k that maximises the product of the likelihoods Λ_{ij} .

$$\prod_{\{e_m \in E_k\}} \Lambda_{e_m} \quad (3.4)$$

which is equivalent to maximising the sum of the log-likelihoods.

$$\sum_{\{e_m \in E_k\}} \ln \Lambda_{e_m} \quad (3.5)$$

or minimising the sum of normalised distances N_{ij} .

$$\sum_{\{e_m \in E_k\}} N_{e_m} \quad (3.6)$$

The method generally used to maximise the log-likelihood sum while ensuring one-to-one assignment is *maximum-weight bipartite graph matching* [35]. Basic renditions of this method assume the bipartite graph is complete (any observations might validly be assigned to any

target), but extensions have been proposed to cater for the existence of false alarms (spurious observations) and non-detections (non-observed targets) [34]. However, models of false alarm probabilities are application specific and often of little value so, if the bipartite graph is incomplete, it is generally preferable to select the optimal assignment set only from the E_k 's of equal maximum size.

Notice that, while optimal assignment uses the injective mapping constraint to prevent conflicting associations, it does not use the concurrent existence of the observation set to influence the value of the association likelihoods. These likelihoods are still calculated on an individual (isolated) basis. An alternative method is proposed in [134], which adjusts the individual assignment probabilities as a batch event, and produces a modified set of likelihoods called the *joint assignment matrix* (JAM). This algorithm introduces the mapping constraint into the likelihood calculation so that the likelihood of a particular observation-to-target pairing is “the sum of the probabilities of all [sets of] assignments containing the pair, normalised by the sum of the probabilities of all assignments” [134, page 134].

$$\Lambda'_{e_m} = \frac{\sum_{\{E_k | \exists e_m \in E_k\}} \prod_{\{\forall e_i \in E_k\}} \Lambda_{e_i}}{\sum_{\{\forall E_k\}} \prod_{\{\forall e_i \in E_k\}} \Lambda_{e_i}} \quad (3.7)$$

The resulting constrained association likelihoods more accurately reflect the combined influence of the observation set, and can be used to obtain better optimal assignment results when substituted into Equation 3.4.

Extensions to PDA and MHT have also been proposed to incorporate the injective mapping constraint. The multiple target version of PDA, *joint probabilistic data association* (JPDA), again utilises all valid associations in a weighted average update but is modified to account for some observations having non-unique correspondence possibilities. The injective mapping constraint means that both JPDA and MHT operate with sets of mutually compatible association pairs, rather than with individual assignment pairs. For the sake of efficiency, the number of association sets is usually limited to the k most likely sets of hypotheses, which are found using an extension of optimal assignment [35].

3.2 Batch Data Association For Correlated Feature Sets

Stochastic SLAM is a form of multiple target tracking problem where the targets are known to have zero relative motion. This is a specific case of a more general target tracking category where the target set (or the observation set) is internally correlated. However, none of the traditional data association methods exploit these correlations to reduce ambiguity. This section presents methods for batch data association that capitalise on the correlations present within a set of features.

Data association requires the transformation of the two data sets into the same coordinate space (usually observation space). If either data set is internally correlated when they are transformed to a common coordinate space, then the batch data association methods described in this section are able to produce better results than is possible with individual validation gating.

Caveat. It is important to make a distinction between ambiguity reduction and ambiguity management. The NIS gate is an ambiguity reduction mechanism, as it constrains the number of observation-to-target associations according to a specified probability of acceptance. (Recall that Gaussians have infinite tails and, without the gate, an observation might correspond to any target.) On the other hand, the methods described in the previous section—nearest neighbour, JPDA, MHT—are ambiguity management techniques, as they attempt to resolve any conflicting data associations that remain after gating. Thus, ambiguity reduction implies reducing the number of association possibilities, and ambiguity management implies determining how to resolve ambiguous association possibilities. The batch data association methods presented in this section fall into the category of ambiguity reduction. They are essentially multiple-target versions of the NIS gate that enforce stronger constraints by incorporating additional correlation information. This results in fewer possible observation-to-target correspondences given the same probability of acceptance. However, ambiguous associations may remain and, for these, the traditional ambiguity management techniques must still be applied (e.g., see Section 3.3.7).

3.2.1 Joint Compatibility Branch and Bound

The *joint compatibility branch and bound* (JCBB) algorithm [105, 106] generates tentative sets of associations and searches for the largest set that satisfies joint compatibility. The search is performed by incrementally constructing an “interpretation tree” (see [63]) of the solution space, which enables efficient search space pruning.

For a given set of association pairs, joint compatibility is determined by calculating a single joint NIS gate. The benefit of joint compatibility is that it preserves the correlation information within the set of observations and predicted observations. Consider a set of observations $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_n]^T$ with covariance model \mathbf{R} and a set of target estimates $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_m]^T$ with covariance \mathbf{P} . A tentative set of associations $E_k = \{e_1, \dots, e_j\}$ is chosen, subject to the individual validation and injective mapping constraints. Let the association pair for e_i be denoted \mathbf{z}_{e_i} and \mathbf{x}_{e_i} , such that the joint observation is given by

$$\mathbf{z}_{E_k} = [\mathbf{z}_{e_1}, \dots, \mathbf{z}_{e_j}]^T \quad (3.8)$$

with covariance \mathbf{R}_{E_k} , and the joint predicted observation is as follows.

$$\hat{\mathbf{z}}_{E_k} = \mathbf{h}_{E_k}(\hat{\mathbf{x}}) = \begin{bmatrix} \mathbf{h}_{e_1}(\hat{\mathbf{x}}) \\ \vdots \\ \mathbf{h}_{e_j}(\hat{\mathbf{x}}) \end{bmatrix} \quad (3.9)$$

The joint innovation and innovation covariance are then calculated as

$$\nu_{E_k} = \mathbf{z}_{E_k} - \hat{\mathbf{z}}_{E_k} \quad (3.10)$$

$$\mathbf{S}_{E_k} = \nabla \mathbf{h}_{\mathbf{x}} \mathbf{P} \nabla \mathbf{h}_{\mathbf{x}}^T + \mathbf{R}_{E_k} \quad (3.11)$$

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}} = \left. \frac{\partial \mathbf{h}_{E_k}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}}$. The joint validation gate, therefore, is found to be

$$M_{E_k} = \nu_{E_k}^T \mathbf{S}_{E_k}^{-1} \nu_{E_k} < \gamma_n \quad (3.12)$$

where the value of n is equal to the dimension of the joint innovation vector.

Example 3.2

Joint compatibility for SLAM. At a given instant in time, the SLAM augmented state vector is defined by $\mathbf{x}_a = [\mathbf{x}_v^T, \mathbf{x}_m^T]^T$ where \mathbf{x}_v is the vehicle pose $[x_v, y_v, \phi_v]^T$ and the map contains three point features $\mathbf{x}_m = [x_1, y_1, x_2, y_2, x_3, y_3]^T$. This SLAM estimate is denoted $\hat{\mathbf{x}}_a$ with covariance \mathbf{P}_a . In this moment, a set of range-bearing observations $\mathbf{z} = [r_1, \theta_1, r_2, \theta_2, r_3, \theta_3]^T$ is obtained. These measurements are modelled as uncorrelated with covariance

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_3 \end{bmatrix} \quad \text{where } \mathbf{R}_i = \begin{bmatrix} \sigma_{r_i}^2 & \sigma_{r_i\theta_i}^2 \\ \sigma_{r_i\theta_i}^2 & \sigma_{\theta_i}^2 \end{bmatrix}$$

A hypothetical association set $E_k = \{(\mathbf{z}_1, \mathbf{x}_2), (\mathbf{z}_3, \mathbf{x}_1)\}$ is chosen, and so the tentative joint observation is given by $\mathbf{z}_{E_k} = [r_1, \theta_1, r_3, \theta_3]^T$ with covariance

$$\mathbf{R}_{E_k} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_3 \end{bmatrix}$$

and the joint predicted observation is

$$\hat{\mathbf{z}}_{E_k} = \mathbf{h}_{E_k}(\hat{\mathbf{x}}_a) = \begin{bmatrix} \mathbf{h}_2(\hat{\mathbf{x}}_a) \\ \mathbf{h}_1(\hat{\mathbf{x}}_a) \end{bmatrix} = \begin{bmatrix} \sqrt{(\hat{x}_2 - \hat{x}_v)^2 + (\hat{y}_2 - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_2 - \hat{y}_v}{\hat{x}_2 - \hat{x}_v}\right) - \hat{\phi}_v \\ \sqrt{(\hat{x}_1 - \hat{x}_v)^2 + (\hat{y}_1 - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_1 - \hat{y}_v}{\hat{x}_1 - \hat{x}_v}\right) - \hat{\phi}_v \end{bmatrix}$$

The joint validation gate can then be calculated trivially from Equations 3.10 to 3.12. The χ^2 threshold for a 95% acceptance probability of true association sets is $\gamma_4 = 9.5$.

The solution space for joint compatibility is depicted as an interpretation tree, where each level of the tree defines the set of possible associations for a particular observation and each descending path from the tree root represents a set of injective associations. Continuing from the example in Figure 3.4, the interpretation tree is pictured in Figure 3.5. The branch-and-bound algorithm in [106] constructs the interpretation tree incrementally using depth-first search with a maximum likelihood branching heuristic.³ At each level, a new node is added to the joint association set and tested for joint NIS. If a node on a given level fails the joint validation test, then its child nodes are also invalid and need not be searched. Also, if a descending path has insufficient non-null nodes to produce an association set greater than or equal to the current maximum, it is not searched. The concept of JCBB is illustrated in Figure 3.5 where the search is marked in bold (with the leftmost branch being searched first).

In [106], the maximum set of valid joint associations is assumed to be the correct data association hypothesis. However, the process of searching through the interpretation tree

³The branching heuristic used in [106] is simply the minimum joint NIS. However, a more correct likelihood measure is the maximum joint likelihood Λ_{E_k} , or minimum joint normalised distance N_{E_k} , which may be obtained by substituting ν_{E_k} and \mathbf{S}_{E_k} into Equation 3.2 or Equation 3.3, respectively.

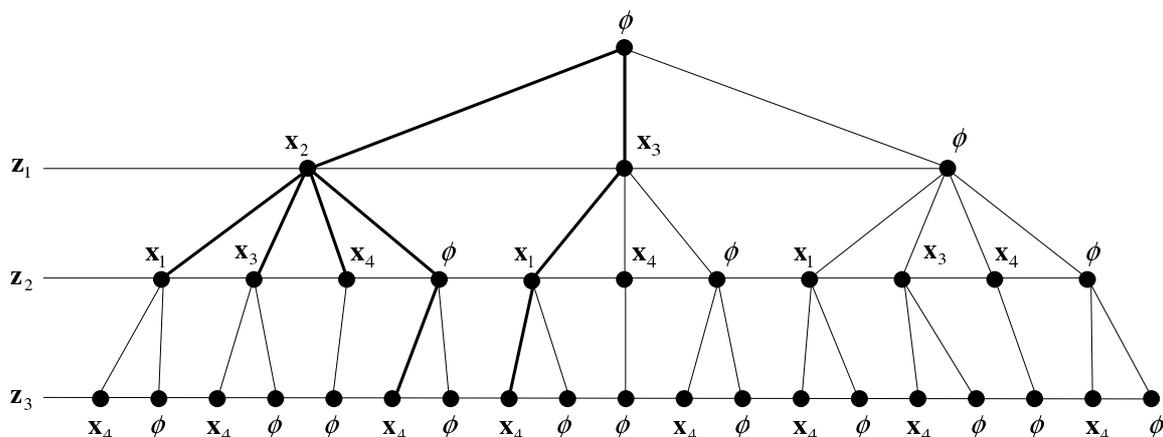


Figure 3.5: Interpretation tree. This figure shows the entire joint compatibility solution space for the assignment example from Figure 3.4. Each level of the tree pertains to a particular observation and each node represents an observation-to-target association (these include possible non-assignments). Each descending path represents an alternative joint association hypothesis. An example maximum joint compatibility branch-and-bound search is shown in bold, with the best result being $E_k = \{(z_1, x_3), (z_2, x_1), (z_3, x_4)\}$.

generates many alternative joint compatible association sets of maximum, or near maximum, size. The coexistence of these alternatives embodies batch data association ambiguity, which can be resolved using traditional ambiguity management methods. For instance, the nearest-neighbour association set is the set with maximum joint likelihood Λ_{E_k} . Note, however, that the interpretation of joint likelihood is complicated by the different sizes $|E_k|$ of the various candidate sets. This is the same problem as discussed in Section 3.1.3 regarding optimal assignment. If the joint likelihoods of unequal sized assignment sets are to be compared, then probability models for false alarms, target non-detections, and new target detections are required; and it is generally difficult to reasonably model these factors. Thus, as with optimal assignment, the best option is usually to select the maximum likelihood result from the set of valid association sets of maximum size.

3.2.2 Maximum Common Subgraph

The *maximum common subgraph* (MCS) batch data association algorithm [5, 4] is a graph-theoretic approach to finding sets of jointly compatible associations.⁴ Each data set is described by a complete graph (i.e., where each node is connected to every other node) such that the nodes represent features, defined by feature type and characteristics, and

⁴MCS-based data association was first proposed in the early 1980s as an object recognition tool [16, 72], but was not previously applied to the problems of multiple target tracking or robot localisation. More recently, a *pair-wise constraint* interpretation tree algorithm has been applied to mobile robot localisation [104, 26], which produces equivalent association results to MCS (including the case where *a priori* pose information is unavailable). However, the branch and bound search of this algorithm is considerably less efficient than a maximum clique implementation of MCS.

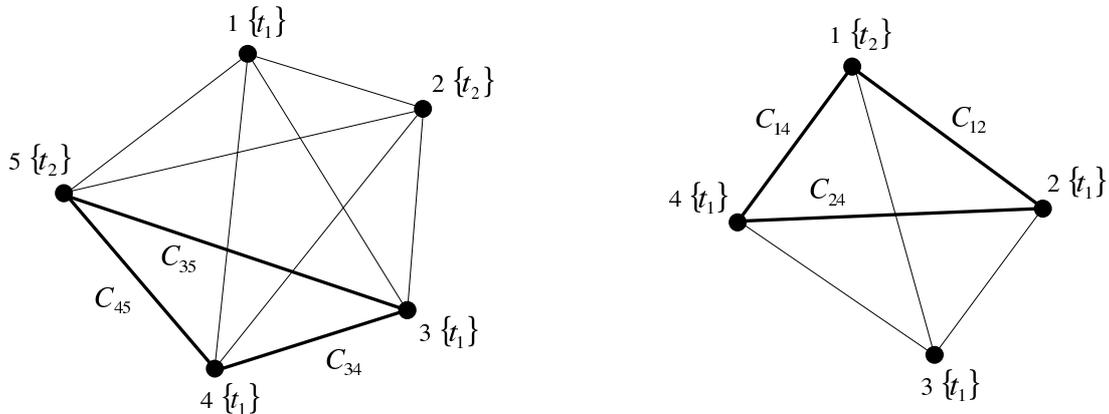


Figure 3.6: Common subgraphs. The two data sets form complete graphs where the nodes are defined by the feature types and the edges are the constraints between features. The common subgraphs shown result from the matching of constraints $\{C_{45}, C_{34}, C_{35}\}$ from the first graph to constraints $\{C_{12}, C_{24}, C_{14}\}$ of the second (along with matching node types).

the edges define pair-wise relational constraints between features. For example, a node might represent a point landmark and an edge the distance between two points. In this thesis, these edges are termed *relative constraints*. The MCS algorithm compares the two feature graphs and finds the maximum complete subgraphs that possess compatible nodes and edges (e.g. see Figure 3.6). Basically, this operation involves matching nodes and edges to find matching subgraphs (see [136, 98, 120, 28] for MCS search algorithms).

The MCS data association algorithm presented in [5] operates purely through matching relative constraints and feature types, which has the effect of enabling batch data association without *a priori* knowledge of the relative pose between the data sets.⁵ However, when pose information is available, a greater degree of association constraint can be achieved through its application. In [4] *pose constrained MCS* is introduced, which uses the *a priori* relative pose information to constrain feature associations between the two data sets (in the same manner as traditional target tracking). Thus, the compatibility of nodes between subgraphs becomes constrained by the proximity of features from one data set to the predicted locations of features from the other, in addition to compatibility of feature type and characteristics. In this thesis, constraints on node compatibility between data sets are collectively termed *absolute constraints*.

The main advantages of the MCS data association method are that the constraints are pair-wise, may be of arbitrary type, and may be incomplete. For example, the relative constraint between two points might be distance, while the relative constraint between two line features might be the subtended angle. Similarly, an absolute constraint might be the type *circle* with radius r . Incomplete constraint information means that edge and node matching between the two graphs may be partially unrestricted. That is, an unknown relative constraint (or unconstrained edge) is compatible with all other edges, and an unknown

⁵For the robot localisation problem, where one data set is the map and the other is the observation scan, the “relative pose” between the two data sets is simply the robot pose with respect to the map coordinate frame. Thus, not having *a priori* knowledge of the relative pose means the robot location is initially unknown.

absolute constraint (or unconstrained node) may correspond to any other node. The effect of incomplete information is most apparent in two extreme cases, where there are only relative constraints or only absolute constraints. In the former, MCS data association is determined purely by relational similarity (this is basically the algorithm presented in [5]). The latter case is simply individual assignment limited by the injective mapping constraint, and has similar behaviour to the optimal assignment algorithm.

MCS and JCBB are similar in that they both utilise all the available correlation information to perform batch data association. JCBB does this by forming association hypotheses and searching an interpretation tree. MCS instigates a two-stage process of (i) compiling pair-wise constraints and (ii) searching for the largest compatible association set. The relative advantages of these two approaches, when given an *a priori* estimate of the relative pose, have not been properly investigated. A probable outcome will be that JCBB is simpler to implement while MCS offers more flexible constraint options. However, MCS offers one significant advantage in its ability to provide a most likely batch association set when the relative pose is initially unknown or very uncertain.

3.3 Combined Constraint Data Association

This section presents a development of the MCS batch data association algorithm called *combined constraint data association* (CCDA). The CCDA algorithm integrates the application of relative and absolute constraints into a unified EKF framework.

The fundamental data structure of the CCDA algorithm is the correspondence graph (CG) [8], which represents valid associations between the two data sets (see Figure 3.7). Complete subgraphs (or cliques) within the CG indicate mutual association compatibility and, by performing maximum clique search, the largest joint compatible association set may be found.⁶

Construction of the CG is performed through the application of relative and absolute constraints. The nodes of the CG indicate individual association compatibility and are determined by absolute constraints (e.g., node 1 is valid if feature a_1 may correspond to b_1). The edges of the CG indicate joint compatibility of the connected nodes and are determined by relative constraints (e.g., if the relative constraints between $\{a_1, a_2\}$ and $\{b_1, b_2\}$ match, then node 1 is connected to node 5).

Explicit examples of relative and absolute constraints are presented in this section, for the range-bearing SLAM problem with point location landmarks. These constraints are statistical thresholds based on the NIS validation gate. Application of more general constraint forms (including non-statistical constraints such as feature type) can be inferred in light of these examples.

⁶Note, as mentioned previously, it is not possible to compare association sets of *different sizes* without models of non-detection, false-detection, etc. Here, as elsewhere [14, 106], the set with the greatest *number* of compatible associations is assumed to represent the maximum likelihood association set—hence the search for the maximum clique. In reality, each clique implies an alternative possible association set and resolution of these ambiguities is addressed in Section 3.3.7.

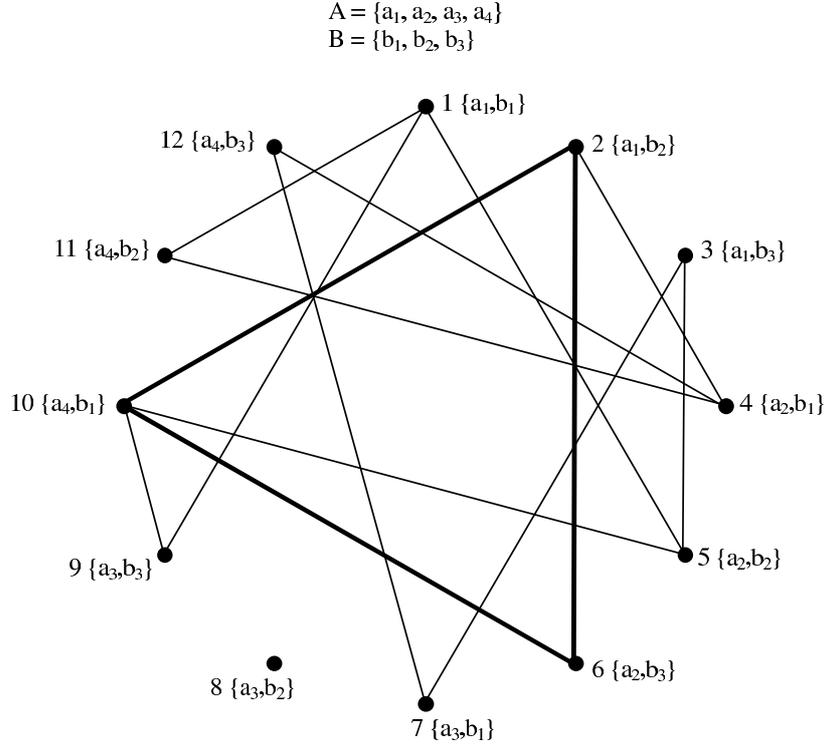


Figure 3.7: Correspondence graph. The (individually) possible associations between feature sets, A and B , are depicted as nodes. Since, in this example, no absolute constraints exist, all combinations of association pairs are valid. The edges of the CG define which pairings are mutually compatible, dependent on injective mapping and relative constraints. The maximum clique of this graph is shown in bold.

3.3.1 Calculation of Relative and Absolute Constraints

In the previous discussions of target tracking data association, the NIS was calculated in range-bearing observation space. However, in this section, the set of observations are transformed to Cartesian space, with the observer as the coordinate origin. Since each observation \mathbf{z}_i is assumed uncorrelated to the others, the polar-to-Cartesian transformation can be performed individually.

$$\mathbf{z}_{C_i} = \mathbf{f}(\mathbf{z}_i) = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{bmatrix} \quad (3.13)$$

$$\mathbf{R}_{C_i} = \nabla \mathbf{f}_{\mathbf{z}_i} \mathbf{R}_i \nabla \mathbf{f}_{\mathbf{z}_i}^T \quad (3.14)$$

The two data sets, therefore, are the augmented SLAM state vector $\hat{\mathbf{x}}_a = [\hat{\mathbf{x}}_v^T, \hat{\mathbf{x}}_m^T]^T$ with covariance \mathbf{P}_a , and the batch observation vector \mathbf{z}_C with covariance \mathbf{R}_C .

Relative constraints concern each data set separately, so let $\hat{\mathbf{x}}$ with covariance \mathbf{P} represent either the state or observation data. The invariant property between the point

features (\hat{x}_i, \hat{y}_i) and (\hat{x}_j, \hat{y}_j) in $\hat{\mathbf{x}}$ is their relative distance⁷

$$d_{ij}(\hat{\mathbf{x}}) = \sqrt{(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2} \quad (3.15)$$

with scalar variance

$$\sigma_{ij}^2 = \mathbf{P}_d = \nabla d_{\mathbf{x}} \mathbf{P} \nabla d_{\mathbf{x}}^T$$

where the Jacobian $\nabla d_{\mathbf{x}}$ is given by

$$\nabla d_{\mathbf{x}} = \left. \frac{\partial d_{ij}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}} = \begin{bmatrix} 0 & \dots & 0 & \frac{\hat{x}_i - \hat{x}_j}{d_{ij}(\hat{\mathbf{x}})} & \frac{\hat{y}_i - \hat{y}_j}{d_{ij}(\hat{\mathbf{x}})} & 0 & \dots & 0 & -\frac{\hat{x}_i - \hat{x}_j}{d_{ij}(\hat{\mathbf{x}})} & -\frac{\hat{y}_i - \hat{y}_j}{d_{ij}(\hat{\mathbf{x}})} & 0 & \dots & 0 \end{bmatrix}$$

A more efficient form of this calculation can be derived considering the sparseness of the Jacobian. First the relevant portions of the covariance matrix are extracted to produce a reduced form⁸

$$\mathbf{P}_r = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_i} & \mathbf{P}_{\mathbf{x}_i \mathbf{x}_j} \\ \mathbf{P}_{\mathbf{x}_i \mathbf{x}_j}^T & \mathbf{P}_{\mathbf{x}_j} \end{bmatrix} \quad (3.16)$$

The distance variance can now be calculated as

$$\sigma_{ij}^2 = \nabla d_{\mathbf{x}_r} \mathbf{P}_r \nabla d_{\mathbf{x}_r}^T \quad (3.17)$$

where the Jacobian $\nabla d_{\mathbf{x}_r}$ is given by

$$\nabla d_{\mathbf{x}_r} = \begin{bmatrix} \frac{\hat{x}_i - \hat{x}_j}{d_{ij}(\hat{\mathbf{x}})} & \frac{\hat{y}_i - \hat{y}_j}{d_{ij}(\hat{\mathbf{x}})} & -\frac{\hat{x}_i - \hat{x}_j}{d_{ij}(\hat{\mathbf{x}})} & -\frac{\hat{y}_i - \hat{y}_j}{d_{ij}(\hat{\mathbf{x}})} \end{bmatrix} \quad (3.18)$$

As relative constraints make no distinction between the observation and state information, let the two data sets be represented by set A and set B . Let C_{A_i} denote a relative constraint $\{d_{pq}, \sigma_{pq}^2\}$ ($\triangleq \{d_{A_i}, \sigma_{A_i}^2\}$) from set A and C_{B_j} denote a relative constraint $\{d_{rs}, \sigma_{rs}^2\}$ from set B . The constraints C_{A_i} and C_{B_j} are said to match if they satisfy the NIS threshold

$$M_{ij} = \frac{(d_{A_i} - d_{B_j})^2}{\sigma_{A_i}^2 + \sigma_{B_j}^2} < \gamma_1 \quad (3.19)$$

Absolute constraints determine individual compatibility across the two data sets. For the SLAM problem, this is determined by the NIS between an observation \mathbf{z}_{C_i} in \mathbf{z}_C and a feature \mathbf{x}_j in \mathbf{x}_a . The predicted observation is modelled as

$$\hat{\mathbf{z}}_{C_j} = \mathbf{h}_j(\hat{\mathbf{x}}_a) = \begin{bmatrix} (\hat{x}_j - \hat{x}_v) \cos \hat{\phi}_v + (\hat{y}_j - \hat{y}_v) \sin \hat{\phi}_v \\ -(\hat{x}_j - \hat{x}_v) \sin \hat{\phi}_v + (\hat{y}_j - \hat{y}_v) \cos \hat{\phi}_v \end{bmatrix} \quad (3.20)$$

The innovation and innovation covariance between observation and predicted observation are as follows.

$$\nu_{ij} = \mathbf{z}_{C_i} - \mathbf{h}_j(\hat{\mathbf{x}}_a) \quad (3.21)$$

$$\mathbf{S}_{ij} = \nabla \mathbf{h}_{\mathbf{x}_a} \mathbf{P}_a \nabla \mathbf{h}_{\mathbf{x}_a}^T + \mathbf{R}_{C_i} \quad (3.22)$$

⁷An alternative relative constraint between $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ is $\mathbf{d}_{ij}(\hat{\mathbf{x}}) = [\hat{x}_i - \hat{x}_j, \hat{y}_i - \hat{y}_j]^T$, which is likely to give better results since it is a *linear* constraint model. The NIS gate in Equation 3.19 would be γ_2 .

⁸Notice that the cross-correlation terms $\mathbf{P}_{\mathbf{x}_i \mathbf{x}_j}$ are utilised in the calculation of relative constraints. It is this information that makes CCDA more constrained than non-batch methods, like optimal assignment.

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_a}$ is given by

$$\nabla \mathbf{h}_{\mathbf{x}_a} = \left. \frac{\partial \mathbf{h}_j}{\partial \mathbf{x}_a} \right|_{\hat{\mathbf{x}}_a} = \begin{bmatrix} -\cos \hat{\phi}_v & -\sin \hat{\phi}_v & -(\hat{x}_j - \hat{x}_v) \sin \hat{\phi}_v + (\hat{y}_j - \hat{y}_v) \cos \hat{\phi}_v \\ \sin \hat{\phi}_v & -\cos \hat{\phi}_v & -(\hat{x}_j - \hat{x}_v) \cos \hat{\phi}_v - (\hat{y}_j - \hat{y}_v) \sin \hat{\phi}_v \\ 0 & \dots & 0 & \cos \hat{\phi}_v & \sin \hat{\phi}_v & 0 & \dots & 0 \\ 0 & \dots & 0 & -\sin \hat{\phi}_v & \cos \hat{\phi}_v & 0 & \dots & 0 \end{bmatrix}$$

As with the relative constraint calculations, the sparse nature of the Jacobian permits more efficient calculation of the innovation covariance by extracting a reduced form of the state covariance matrix.

$$\mathbf{S}_{ij} = \nabla \mathbf{h}_{\mathbf{x}_r} \mathbf{P}_r \nabla \mathbf{h}_{\mathbf{x}_r}^T + \mathbf{R}_{C_i} \quad (3.23)$$

$$\mathbf{P}_r = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{v\mathbf{x}_j} \\ \mathbf{P}_{v\mathbf{x}_j}^T & \mathbf{P}_{\mathbf{x}_j} \end{bmatrix} \quad (3.24)$$

$$\nabla \mathbf{h}_{\mathbf{x}_r} = \begin{bmatrix} -\cos \hat{\phi}_v & -\sin \hat{\phi}_v & -(\hat{x}_j - \hat{x}_v) \sin \hat{\phi}_v + (\hat{y}_j - \hat{y}_v) \cos \hat{\phi}_v & \cos \hat{\phi}_v & \sin \hat{\phi}_v \\ \sin \hat{\phi}_v & -\cos \hat{\phi}_v & -(\hat{x}_j - \hat{x}_v) \cos \hat{\phi}_v - (\hat{y}_j - \hat{y}_v) \sin \hat{\phi}_v & -\sin \hat{\phi}_v & \cos \hat{\phi}_v \end{bmatrix} \quad (3.25)$$

The absolute constraint of NIS validity is therefore calculated as

$$M_{ij} = \nu_{ij}^T \mathbf{S}_{ij}^{-1} \nu_{ij} < \gamma_2 \quad (3.26)$$

3.3.2 Set and Graph Terminology

The pseudocode modules in this section are described using set and graph notation. The meanings of these notations in the context of the CCDA algorithm are listed below.

A set S is a collection of elements $\{s_1, \dots, s_n\}$ with the number of elements given by $|S|$. An empty set is denoted \emptyset . The existence of element s_i in set S is denoted $s_i \in S$. Two sets are combined by their union $S_3 \leftarrow S_1 \cup S_2$. Thus, new elements may be added to a set as $S \leftarrow S \cup \{s_{new}\}$. The set difference $S_3 \leftarrow S_1 \setminus S_2$ assigns to S_3 the elements in S_1 that are not in S_2 . This operation can be used to remove an element from a set as $S \leftarrow S \setminus \{s_i\}$. The intersection of two sets $S_3 \leftarrow S_1 \cap S_2$ defines the set of elements common to both sets. Most of the sets specified in the code modules below are ordered sets (also known as vectors or arrays) as they require a specific iteration order. Adding new elements to these sets by the union operator \cup implies addition to the back of the set. Similarly, the difference operator \setminus removes elements from the back of an ordered set.

An undirected unweighted graph $G = \{V, E\}$ is defined as a set of nodes (or vertices) $V = \{v_1, \dots, v_n\}$ and a set of edges $E = \{e_1, \dots, e_m\}$ where each edge $e_k = \{v_i, v_j\}$ represents a connection between two nodes. Two nodes v_i and v_j are *adjacent* if there exists an $e_k = \{v_i, v_j\} \in E$. The set of nodes adjacent to node v_i is given by $adj(v_i)$ which represents the set of all $v_j \in V$ for which there exists $e_k = \{v_i, v_j\} \in E$. Each node $v_i = \{x, adj(v_i)\}$ represents a stored item x and links to its adjacent nodes. A graph $G = \{V, E\}$ is *complete* if all its vertices are adjacent. A clique is a complete subgraph of G . A clique $G_M \subset G$ is *maximal* if no additional node in G is adjacent to all the nodes contained in G_M . The largest maximal clique in G is termed the maximum clique (and there may exist several maximum cliques).

3.3.3 Correspondence Graph with Unknown Vehicle Pose

The construction of the CG is presented here for the case where there exist no absolute constraints.⁹ This situation is encountered with pose initialisation in a (partially) known map, and recovery from localisation failure where the robot has become lost.

Data association based solely on relative constraints operates by finding subsets of map features that possess similar geometric configuration to subsets of the observed feature set. An example of CCDA with only relative constraints is shown in Figure 3.8. Both figures represent the same two feature sets extracted from real scanning laser data but the bottom figure depicts a false association set due to environmental symmetry. In both cases, the relative pose between the two feature sets is calculated subsequently using the method in Appendix C.3. Usually, as in this example, the correct association set is larger than for incorrect hypotheses, but this may not always be the case and, in general, symmetry-based ambiguities must be resolved using MHT.

The CG construction algorithm `CORRESPONDENCEGRAPH(A, B)` in Algorithm 3.1 takes as input two feature sets $A = \{\hat{\mathbf{x}}_A, \mathbf{P}_A\}$ and $B = \{\hat{\mathbf{x}}_B, \mathbf{P}_B\}$. The set of nodes for the unconnected CG are generated via the function `MAKENODESET(m, n, G)` (see Algorithm 3.2). Each node represents a possible association and, since there are no absolute constraints, this means that mn nodes are created. The node set V is ordered such that each node $v_k = \{a_i, b_j\}$ can be found as $k = (a_i - 1)n + b_j$.

The generation of relative constraints for each data set is performed via the function `MAKERELATIVECONSTRAINTS(F)` in Algorithm 3.3. Given a set of features $F = \{\hat{\mathbf{x}}, \mathbf{P}\}$, a constraint $C_k = \{i, j, d_{ij}, \sigma_{ij}^2\}$ is calculated for each combination of feature pairings. In Algorithm 3.1, the second constraint set C_B is subsequently ordered via a sorting function `SORT(C_B)`, which arranges the elements $C_{B_k} \in C_B$ so that the distance estimates d_{B_k} are in ascending order. This enables efficient constraint matching between the two data sets.

CG edges are generated by matching the two relative constraint sets C_A and C_B as shown in Figure 3.9. To permit search efficiency, the matching region is restricted by a fixed bound Δ_d since, without this restriction, the constraint C_{A_i} would have to be checked against all $C_{B_k} \in C_B$. The consequence of Δ_d is that some matching constraints in C_B are not tested. This is a reasonable tradeoff, in practice, as the efficiency gained is substantial and association possibilities rejected by the bounded search tend to be unlikely.

The details of constraint matching are provided in Algorithm 3.4 where the function `MATCHCONSTRAINTS(C_A, C_B, G)` takes as input the two relative constraint sets and the unconnected graph $G = \{V, \emptyset\}$. For each constraint $C_{A_i} \in C_A$, the constraints $\{C_{B_j}\} \subset C_B$, satisfying $d_{A_i} - \Delta_d \leq \{d_{B_j}\} < d_{A_i} + \Delta_d$, are checked as follows. First, a binary search¹⁰ is performed using the function `UPPERBOUND($C_B, d_{A_i} - \Delta_d$)` which returns the index j of the first constraint in C_B with $d_{B_j} \geq d_{A_i} - \Delta_d$. Each ensuing C_{B_j} satisfying $d_{B_j} < d_{A_i} + \Delta_d$ is then compared using the function `CHECKCONSTRAINT(C_{A_i}, C_{B_j})`, shown in Algorithm 3.5, which calculates the NIS gate given in Equation 3.19.

Let C_{A_i} be the relative constraint between feature pair $\{a_r, a_s\}$ and C_{B_j} the pair $\{b_p, b_q\}$.

⁹Features are assumed to be identical point targets in this discussion and absolute constraints based on feature distinctiveness are not considered. However, addition of these constraints, as for pose-based absolute constraints, is straightforward using the method presented in Section 3.3.4.

¹⁰The C++ STL function `upper_bound()` performs the form of binary search task described.

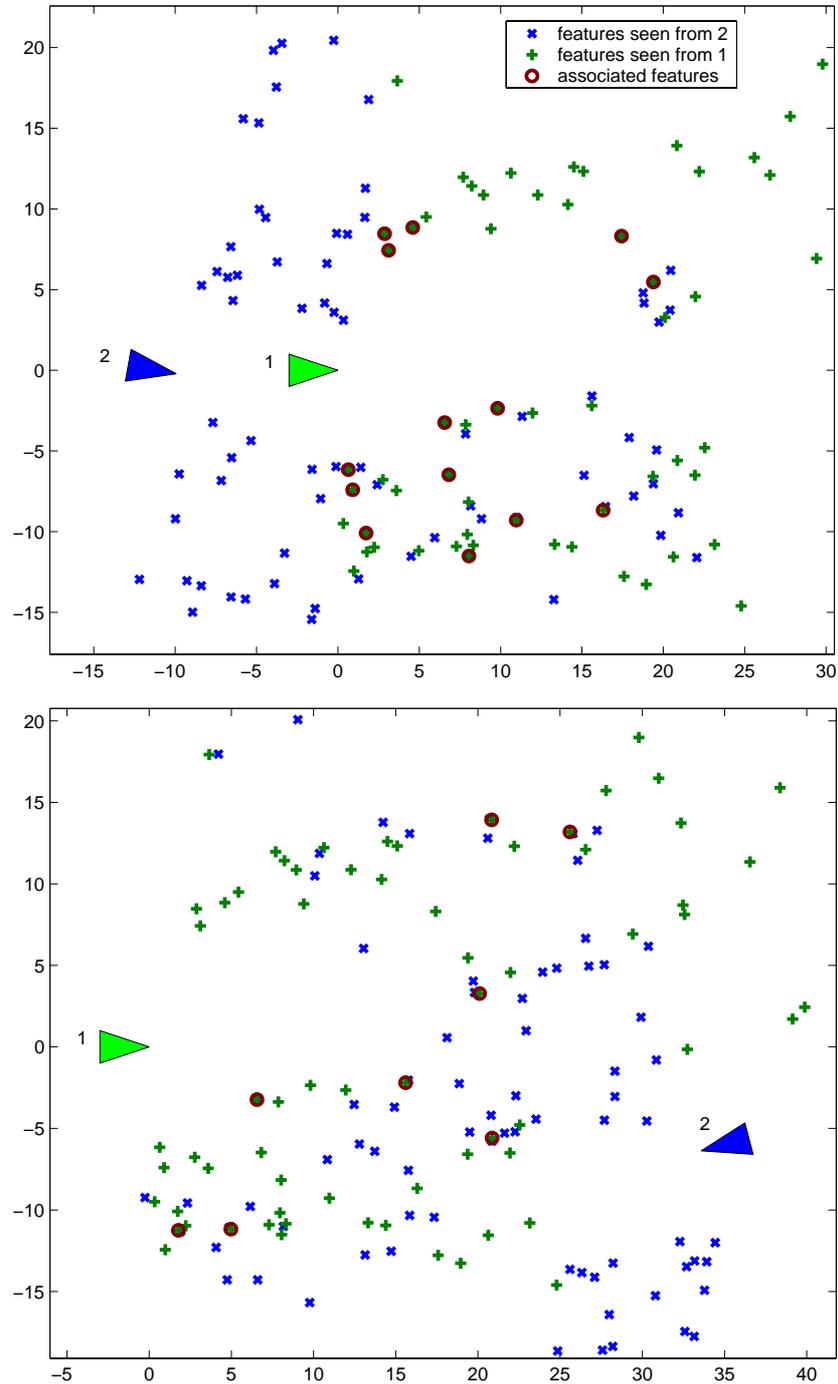


Figure 3.8: CCDA with relative constraints. The above figures (dimensions in metres) show two association possibilities for the same two sets of real data. The relative pose between these data sets is not known *a priori*, and both contain large numbers of spurious features. The location of the observer for each set is indicated by a triangle, and their positions depict the estimated relative pose derived from association. The top figure shows a correct association set and the bottom figure shows an incorrect association set.

Algorithm 3.1: CORRESPONDENCEGRAPH(A, B)

```

 $G \leftarrow \{V = \emptyset, E = \emptyset\}$ 
 $G \leftarrow \text{MAKENODESET}(|A|, |B|, G)$ 
 $C_A \leftarrow \text{MAKERELATIVECONSTRAINTS}(A)$ 
 $C_B \leftarrow \text{MAKERELATIVECONSTRAINTS}(B)$ 
 $\text{SORT}(C_B)$ 
 $G \leftarrow \text{MATCHCONSTRAINTS}(C_A, C_B, G)$ 
return ( $G$ )

```

Algorithm 3.2: MAKENODESET(m, n, G)

```

for  $i \leftarrow 1$  to  $m$ 
  do  $\left\{ \begin{array}{l} \text{for } j \leftarrow 1 \text{ to } n \\ \text{do } V \leftarrow V \cup \{i, j\} \end{array} \right.$ 
return ( $G$ )

```

Algorithm 3.3: MAKERELATIVECONSTRAINTS(F)

```

 $C \leftarrow \emptyset$ 
for  $i \leftarrow 1$  to  $|F|$ 
   $\left\{ \begin{array}{l} \text{for } j \leftarrow i + 1 \text{ to } |F| \\ \text{do } \left\{ \begin{array}{l} d_{ij} \leftarrow \text{Equation 3.15} \\ \sigma_{ij}^2 \leftarrow \text{Equation 3.17} \\ C \leftarrow C \cup \{i, j, d_{ij}, \sigma_{ij}^2\} \end{array} \right. \end{array} \right.$ 
return ( $C$ )

```

Algorithm 3.4: MATCHCONSTRAINTS(C_A, C_B, G)

```

for  $i \leftarrow 1$  to  $|C_A|$ 
  do  $\left\{ \begin{array}{l} d \leftarrow d_{A_i} \in C_{A_i} \\ j \leftarrow \text{UPPERBOUND}(C_B, d - \Delta_d) \\ \text{while } (d_{B_j} \in C_{B_j}) < (d + \Delta_d) \\ \text{do } \left\{ \begin{array}{l} \text{if } \text{CHECKCONSTRAINT}(C_{A_i}, C_{B_j}) \\ \text{then } G \leftarrow \text{MAKEEDGES}(C_{A_i}, C_{B_j}, G) \\ j \leftarrow j + 1 \end{array} \right. \end{array} \right.$ 
return ( $G$ )

```

Algorithm 3.5: CHECKCONSTRAINT(C_{A_i}, C_{B_j})

$$M_{ij} \leftarrow \frac{(d_{A_i} - d_{B_j})^2}{\sigma_{A_i}^2 + \sigma_{B_j}^2}$$

return ($M_{ij} < \gamma_1$)

Algorithm 3.6: MAKEEDGES(C_{A_i}, C_{B_j}, G)

```

 $k_1 \leftarrow \text{NODEINDEX}(a_r \in C_{A_i}, b_p \in C_{B_j})$ 
 $k_2 \leftarrow \text{NODEINDEX}(a_s \in C_{A_i}, b_q \in C_{B_j})$ 
 $E \leftarrow E \cup \{v_{k_1}, v_{k_2}\}$ 
 $k_1 \leftarrow \text{NODEINDEX}(a_r \in C_{A_i}, b_q \in C_{B_j})$ 
 $k_2 \leftarrow \text{NODEINDEX}(a_s \in C_{A_i}, b_p \in C_{B_j})$ 
 $E \leftarrow E \cup \{v_{k_1}, v_{k_2}\}$ 
return ( $G$ )

```

Algorithm 3.7: NODEINDEX(a_i, b_j)

```

 $k \leftarrow (a_i - 1)n + b_j$ 
return ( $k$ )

```

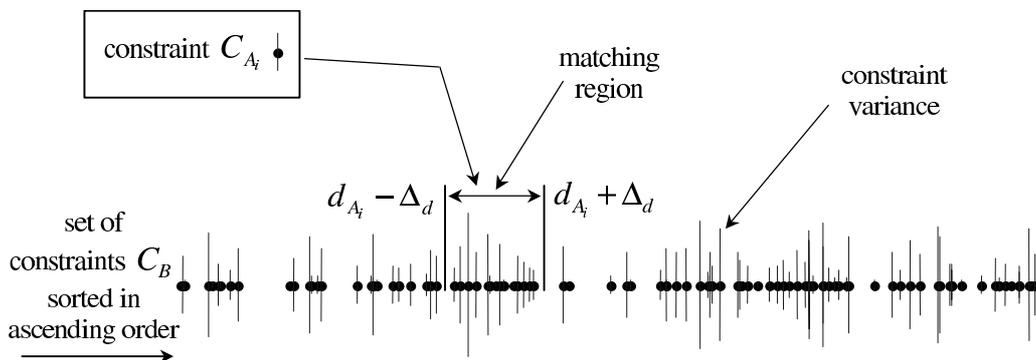


Figure 3.9: Relative constraint matching. The elements of constraint set C_B are sorted in ascending order according to their distance values d_{B_j} (representing the distance between features $\{b_p, b_q\}$). The variance $\sigma_{B_j}^2$ of each constraint is signified here by a vertical line. Matching constraint C_{A_i} to elements of C_B is performed as follows. The first element d_{B_j} greater than $d_{A_i} - \Delta_d$ is found by binary search. A check for match is then performed with each element C_{B_j} possessing d_{B_j} in the region $d_{A_i} - \Delta_d$ to $d_{A_i} + \Delta_d$.

A valid matching means that the associations $\{a_r, b_p\}$ and $\{a_s, b_q\}$ are mutually compatible, and $\{a_r, b_q\}$ and $\{a_s, b_p\}$ are mutually compatible (although, clearly, these two hypotheses are not compatible with each other). This results in two edges being added to the CG via the function $\text{MAKEEDGES}(C_{A_i}, C_{B_j}, G)$ in Algorithm 3.5. The indices of the connected nodes are found in $O(1)$ time using the order of set V as shown in Algorithm 3.7.

The computational complexity of Algorithm 3.4 is not readily apparent because the inner **while** loop is non-deterministic. On average the **while** loop performs approximately k iterations, where k is dependent on n , Δ_d , and the distribution of $\{d_{B_j}\}$ in C_B . An empirical evaluation of k , as a function of n , is shown in Figure 3.10. These measurements are taken from the laser-based dead reckoning experiment presented in Section 3.4 using data from the internal road environment (see Appendix A). From these results, the complexity of k appears to be quadratic with respect to n , for fixed Δ_d . However, over the experimental range of n , this complexity is bounded above by a small-constant *linear* bound $k = 2n$. This is a substantial improvement over the brute force complexity $k = \frac{n^2 - n}{2}$ for comparing C_{A_i} to every $C_{B_j} \in C_B$.

The total complexity of Algorithm 3.1, therefore, is $O(mn + m^2 + n^2 + n^2 \log n^2 + m^2 \log n^2 + km^2)$, which may be simplified to $O(n^2 \log n^2 + km^2)$. If it is assumed that $m \approx n$, this can be further simplified to $O(km^2)$.

3.3.4 Correspondence Graph with Partially Known Vehicle Pose

With the existence of partial knowledge of the observer pose, absolute constraints can be incorporated into the CG construction process. Absolute constraints minimise the problem of environmental symmetry by rejecting associations that fall outside the threshold of observer pose uncertainty. As a result, only local symmetries, that fall within the pose

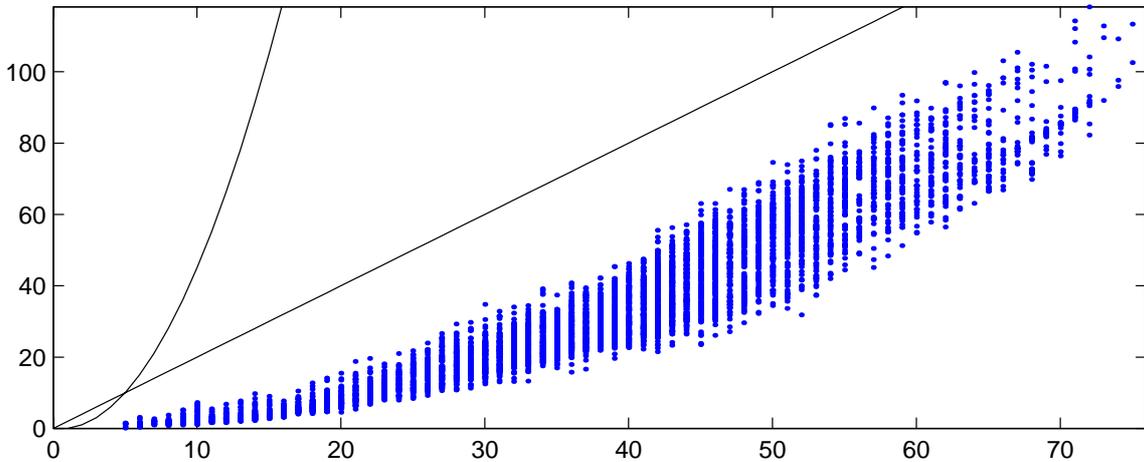


Figure 3.10: Inner loop of CG construction. Empirical measurements for the number of iterations k of the **while** loop in the function $\text{MATCHCONSTRAINTS}(C_A, C_B, G)$ are shown as a function of n , where n is the number of features in set B (and $\Delta_d = 1.0$ metre). The brute force complexity $k = \frac{n^2 - n}{2}$ and the linear bound $k = 2n$ are shown for comparison.

uncertainty region, remain ambiguous.

Two CG construction algorithms are presented in this section. The first applies absolute constraints after each relative constraint match, and is a simple extension to the method in Section 3.3.3. The second applies absolute constraints prior to the relative constraint matching process, and is a very efficient implementation for situations where the vehicle is already reasonably well localised.

Method 1: Post application of absolute constraints

Let the data set A represent the observation data set $\{\mathbf{z}_C, \mathbf{R}_C\}$, and the set B represent the augmented state estimate $\{\hat{\mathbf{x}}_a, \mathbf{P}_a\}$. The following algorithm differs from the algorithm in Section 3.3.3 only in the implementation of the function $\text{MAKEEDGES}(C_{A_i}, C_{B_j}, G)$ (see Algorithm 3.6). The new version of this function checks the node validity for each edge entry and inserts a new edge into the CG only if the nodes satisfy Equation 3.26. For the purpose of this calculation, the function is additionally passed the data sets A and B as shown in Algorithm 3.8. The absolute constraint check is performed in Algorithm 3.9 by the function $\text{CHECKNODE}(a_i, b_j, A, B)$, which takes as input the indices of two features a_i and b_j and the feature sets A and B . It returns a Boolean value indicating whether the absolute constraint for the association $a_i \rightarrow b_j$ is satisfied.

Checking the absolute constraint for each node is an $O(1)$ computation and, if implemented as a lookup table so that M_{ij} for each node is evaluated at most once, Algorithm 3.9 can be performed very efficiently. Thus, the complexity of the correspondence graph construction remains $O(n^2 \log n^2 + km^2)$ as for the unknown observer pose algorithm in the previous section.

```

Algorithm 3.8: MAKEEDGES( $C_{A_i}, C_{B_j}, A, B, G$ )

if CHECKNODE( $a_r, b_p, A, B$ ) and CHECKNODE( $a_s, b_q, A, B$ )
  then  $\begin{cases} k_1 \leftarrow \text{NODEINDEX}(a_r, b_p) \\ k_2 \leftarrow \text{NODEINDEX}(a_s, b_q) \\ E \leftarrow E \cup \{v_{k_1}, v_{k_2}\} \end{cases}$ 
if CHECKNODE( $a_r, b_q, A, B$ ) and CHECKNODE( $a_s, b_p, A, B$ )
  then  $\begin{cases} k_1 \leftarrow \text{NODEINDEX}(a_r, b_q) \\ k_2 \leftarrow \text{NODEINDEX}(a_s, b_p) \\ E \leftarrow E \cup \{v_{k_1}, v_{k_2}\} \end{cases}$ 
return ( $G$ )

```

```

Algorithm 3.9: CHECKNODE( $a_i, b_j, A, B$ )

 $\nu_{ij} \leftarrow$  Equation 3.21
 $\mathbf{S}_{ij} \leftarrow$  Equation 3.23
 $M_{ij} \leftarrow \nu_{ij}^T \mathbf{S}_{ij}^{-1} \nu_{ij}$ 
return ( $M_{ij} < \gamma_2$ )

```

Method 2: Prior application of absolute constraints

If the vehicle pose uncertainty is small, then the set of valid associations can be largely determined from absolute constraints alone. Association set consistency can be subsequently enforced by applying relative constraints to a greatly reduced solution space.

This version of CORRESPONDENCEGRAPH(A, B) is shown in Algorithm 3.10. The set of valid nodes V is generated via the function MAKENODESET(A, B) (see Algorithm 3.11), which applies absolute constraints using the function CHECKNODE(a_i, b_j, A, B) shown previously in Algorithm 3.9. This node generation process takes $O(mn)$ operations.¹¹

The mutual compatibility of each node-pair $\{v_i, v_j\}$ is then tested using the function CHECKCONSTRAINT(v_i, v_j, A, B) in Algorithm 3.12. This function first applies the injective mapping constraint by rejecting nodes $v_i = \{a_r, b_p\}$ and $v_j = \{a_s, b_q\}$ if either $a_r = a_s$ or $b_p = b_q$. Surviving node-pairs are then tested for relative constraint compatibility by comparing the constraints $\{d_{rs}, \sigma_{rs}^2\}$ and $\{d_{pq}, \sigma_{pq}^2\}$. For each compatible node-pair, a connecting edge is added to the CG.

Algorithm 3.10 has computational complexity $O(mn + |V|^2)$. The efficiency of this approach depends on the observer pose uncertainty being small enough that the size of set V is approximately $O(\min\{m, n\})$. That is, the assignments between sets A and B are largely determined by the absolute constraints. If this assumption holds, the overall computational complexity becomes $O(mn)$.

¹¹The process of applying absolute constraints may be improved using more efficient gating procedures, as presented in [32] for example. This would mean Algorithm 3.11 could be performed in $O(m \log n)$ iterations.

Algorithm 3.10: CORRESPONDENCEGRAPH(A, B)

```

 $V \leftarrow \text{MAKENODESET}(A, B)$ 
 $E \leftarrow \emptyset$ 
for  $i \leftarrow 1$  to  $|V|$ 
  do  $\left\{ \begin{array}{l} \text{for } j \leftarrow i + 1 \text{ to } |V| \\ \text{do } \left\{ \begin{array}{l} \text{if } \text{CHECKCONSTRAINT}(v_i, v_j, A, B) \\ \text{then } E \leftarrow E \cup \{v_i, v_j\} \end{array} \right. \end{array} \right.$ 
return ( $G$ )

```

Algorithm 3.11: MAKENODESET(A, B)

```

 $V \leftarrow \emptyset$ 
for  $i \leftarrow 1$  to  $|A|$ 
  do  $\left\{ \begin{array}{l} \text{for } j \leftarrow 1 \text{ to } |B| \\ \text{do } \left\{ \begin{array}{l} \text{if } \text{CHECKNODE}(a_i, b_j, A, B) \\ \text{then } V \leftarrow V \cup \{a_i, b_j\} \end{array} \right. \end{array} \right.$ 
return ( $V$ )

```

Algorithm 3.12: CHECKCONSTRAINT(v_i, v_j, A, B)

```

if ( $a_r \in v_i = a_s \in v_j$ ) or ( $b_p \in v_i = b_q \in v_j$ )
  then return ( false )
 $d_{rs} \leftarrow$  Equation 3.15 for features  $a_r$  and  $a_s$ 
 $\sigma_{rs}^2 \leftarrow$  Equation 3.17 for features  $a_r$  and  $a_s$ 
 $d_{pq} \leftarrow$  Equation 3.15 for features  $b_p$  and  $b_q$ 
 $\sigma_{pq}^2 \leftarrow$  Equation 3.17 for features  $b_p$  and  $b_q$ 
 $M_{ij} \leftarrow \frac{(d_{rs} - d_{pq})^2}{\sigma_{rs}^2 + \sigma_{pq}^2}$ 
return ( $M_{ij} < \gamma_1$ )

```

3.3.5 Maximum Clique Search

Maximum clique search is a standard graph search algorithm, and a great number of techniques may be found in the literature for finding the maximum clique in an undirected unweighted graph (see [17] for a comprehensive survey).

Finding an exact solution to the maximum clique problem in a general graph is thought to be NP complete [78], but two factors make it a feasible algorithm for the CCDA application. First, in typical environments (with normal levels of geometric symmetry), the CG is very sparse and search heuristics favouring nodes of high degree tend to rapidly obtain good results. Second, the CCDA algorithm does not require the exact maximum clique(s) but, rather, the largest maximal clique (or set of maximal cliques) that can be found within a reasonable time period. In fact, an approximate maximum clique search algorithm may be trivially implemented as an *anytime algorithm* [15] where the search runs until a specified time limit and then returns its best result.

Two approximate maximum clique algorithms were empirically trialed for CCDA: a degree-ordered search with a tabu queue [58], and a simple randomised descent search. In most cases the latter method was more efficient and so this version is presented below.

The function `MAXIMUMCLIQUE(G)` (see Algorithm 3.13) is sent the graph $G = \{V, E\}$ as input and returns a clique $M \subseteq G$ that is (approximately) the maximum clique of G .¹² The algorithm uses three global variables: the largest clique found M , a temporary clique T , and a set of nodes S . These variables are shared by the other functions that declare them as global. In Algorithm 3.13, clique search is performed in a loop terminated when a timing function `TIMEUP()` returns `true`. This loop provides the base level of the randomised descent search where a node v_i is chosen at random from $V \in G$, and the node set S is initialised with the adjacency list $adj(v_i)$. The randomised descent search is then performed by recursion of the function `RECURSIVESHARE(n)`.

In Algorithm 3.14, the function `GETRANDOMNODE(V)` chooses a node at random from the node set V (i.e., a node index is selected with uniform¹³ likelihood from the discrete range $[1, |V|]$). Note, the input parameter V is local to this function and does not specify the set $V \in G$ (although $V \in G$ may indeed be passed to this function). For a node v_i , the maximum possible clique size is $|adj(v_i)| + 1$; therefore, a selected node is accepted for searching if the size of its adjacency list is at least as large as the current maximum clique M . If a valid node cannot be found within the final time limit, an empty set is returned.

The function `RECURSIVESHARE(n)` in Algorithm 3.15 is the central process of the maximum clique search operation. Each call of this function indicates the entry to a lower search depth and a new node is added to the accumulating clique stored in T , which represents the current descent path. A set of nodes V that is adjacent to all the nodes currently stored in T is used to perform the next descent level. Provided $|V|$ is large enough to possibly improve the maximum clique size, a node is selected at random from V and another level of `RECURSIVESHARE(n)` is entered. After the completion of the recursive

¹²The cliques are actually represented in Algorithm 3.13 by sets of nodes only rather than as graphs as the nodes implicitly (by definition) form a complete graph.

¹³Rather than simply selecting a node v_i with uniform likelihood, better performance may be possible if the likelihood of selecting node v_i is a function of its degree $|adj(v_i)|$ meaning that more strongly connected nodes are favoured for addition to the clique.

Algorithm 3.13: MAXIMUMCLIQUE(G)

```

global  $M, T, S$ 
 $M \leftarrow \emptyset$ 
 $T \leftarrow \emptyset$ 
while not TIMEUP()
  do  $\begin{cases} v_i \leftarrow \text{GETRANDOMNODE}(V) \\ S \leftarrow \text{adj}(v_i) \\ \text{RECURSIVESHARE}(v_i) \end{cases}$ 
return ( $M$ )

```

Algorithm 3.14: GETRANDOMNODE(V)

```

global  $M$ 
 $n \leftarrow \emptyset$ 
while  $n = \emptyset$  and not TIMEUP()
  do  $\begin{cases} i \leftarrow \mathcal{U}[1, |V|] \\ \text{if } |\text{adj}(v_i)| \geq |M| \\ \text{then } n \leftarrow v_i \end{cases}$ 
return ( $n$ )

```

Algorithm 3.15: RECURSIVESHARE(n)

```

global  $M, T$ 
if  $n = \emptyset$ 
  then return
 $T \leftarrow T \cup n$ 
 $V \leftarrow \text{MUTUALLYADJACENTNODES}(n)$ 
if  $|V| > 0$  and  $|T| + |V| > |M|$ 
  then  $\begin{cases} v_i \leftarrow \text{GETRANDOMNODE}(V) \\ \text{RECURSIVESHARE}(v_i) \end{cases}$ 
if  $|T| > |M|$ 
  then  $M \leftarrow T$ 
 $T \leftarrow T \setminus n$ 

```

Algorithm 3.16: MUTUALLYADJACENTNODES(n)

```

global  $S$ 
 $S \leftarrow S \cap \text{adj}(n)$ 
return ( $S$ )

```

descent, the size of the maximal clique T is compared with the current maximum clique. If T is greater, then it is assigned to M . The node n is subsequently removed from T at the end of the function to indicate the return to a higher level.

The function `MUTUALLYADJACENTNODES(n)` finds the intersection of the sets S and $adj(n)$ and stores the result in S so that, at each level of descent, the nodes in S represent the remaining nodes in the graph G adjacent to all the nodes in the current clique T (but not already stored in T).

This basic implementation finds only a single largest clique but can easily be modified to store the set of all maximal cliques found (a maximal clique is obtained with each descending path). Alternatively, the k largest maximal cliques might be retained. It is possible that the implementation of more sophisticated maximum clique search algorithms (e.g., [10, 54, 28]) may enable more efficient application of the CCDA algorithm, thereby permitting batch associations with larger data sets, but determining the computational advantage of these algorithms for the particular structure of the CG will require an empirical evaluation.

3.3.6 Computational Complexity of the Maximum Clique Search

While maximum clique search may be NP complete for a general graph, it appears to be feasible here due to the structure of the CG. This section presents an empirical study of the CG density and maximum clique search complexity, both with and without absolute constraints.

These experimental results are obtained from the laser-based dead reckoning trials in Section 3.4 using the internal road data set (see Appendix A). CCDA was performed between the feature sets obtained from sequential pairs of laser scans. This test was performed twice over the same ensemble of feature sets: once including absolute constraints and once without (i.e., using relative constraints only).

The structure of the CG is affected by numerous factors. These include the density of observable features (and clutter), their geometric distribution, the measurement (sensor) uncertainty, and the vehicle pose uncertainty. The effect of pose uncertainty (i.e., absolute constraints) on the number of CG nodes is particularly significant. Given moderate absolute constraints, the number of CG nodes is proportional to the number of features in the smaller of the two data sets while, without absolute constraints, the number of nodes is equal to the number of features in one set multiplied by the number of features in the other.

The empirical CG density results, edges versus nodes, are shown in Figure 3.11. In both cases, the number of edges is quadratic in relation to the number of nodes. However, with absolute constraints, the CG is quite dense (between 20% and 60% connected) while, without absolute constraints, the CG density is between 2% and 4%.

In the absence of a theoretical bound for the maximum clique search, the empirical results in Figures 3.12 and 3.13 serve to characterise the expected complexity of the randomised descent algorithm. Figure 3.12 shows the number of descent iterations (i.e., the number of calls to the function `RECURSIVESHARE(n)`) required to find the (exact) maximum clique, versus the number of CG nodes. Because the algorithm is randomised, the computational spread is large; however, the important attribute is the average case as shown.

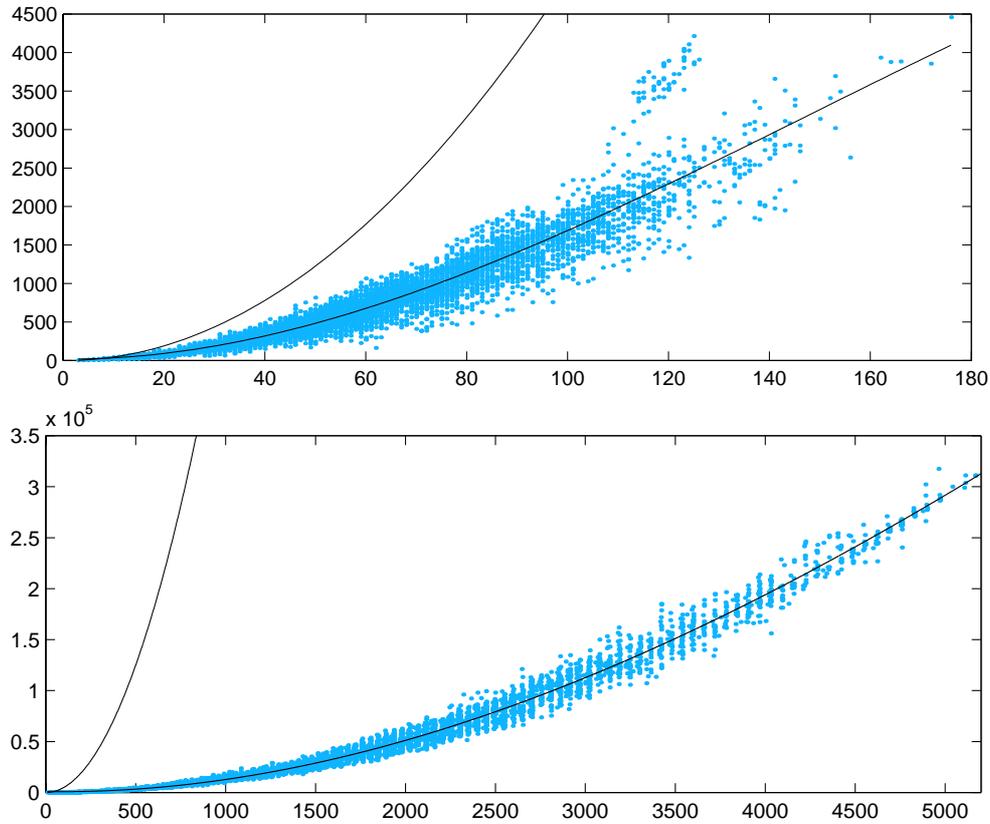


Figure 3.11: Correspondence graph density. These figures show empirical measurements for the number of CG edges versus number of nodes. The top and bottom figures depict the results for CGs with and without absolute constraints, respectively. The two lines in each figure show the average CG density and, for comparison, the density of a complete graph where $e = \frac{n^2 - n}{2}$.

For the first case (with absolute constraints), the number of descent calls is linear with the number of nodes. This is expected since the CG is dense and contains essentially a small number of large cliques. (Note, the apparent drop-off for $n > 140$ is just an artifact of insufficient trials for these larger node quantities.) The second case, without absolute constraints, is particularly interesting as the number of descents achieves a constant bound for large n . It appears that the sparse CG contains relatively few highly connected nodes, and the graph is composed of a small number of large cliques amidst a high proportion of very small cliques.

However, the results in the second case do not tell the entire story for CCDA without absolute constraints. In these experiments, features are matched between sequential laser-scan pairs, meaning that the feature sets are of similar size, and have many features in common. Thus, those CGs with many nodes tend to possess correspondingly large (and distinctive) cliques. A more difficult problem would be associating a small number of observations in a large map without absolute constraints. This would mean a large number

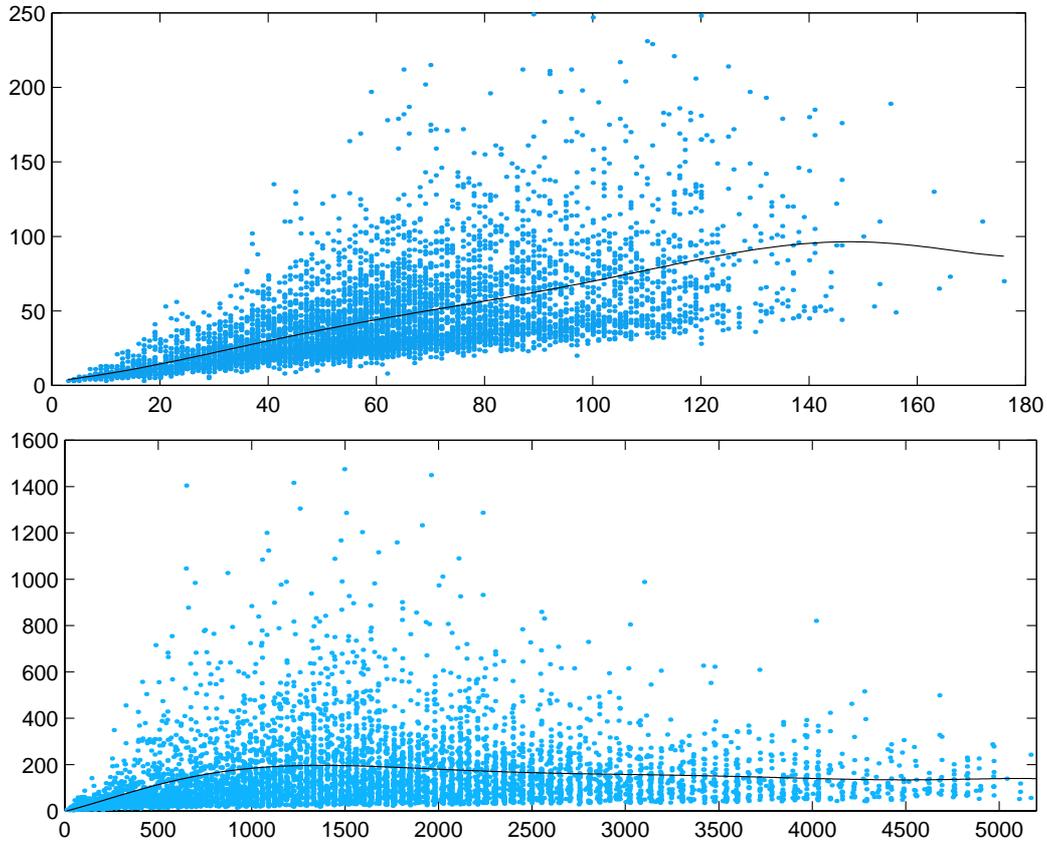


Figure 3.12: Maximum clique search iterations. These figures show the number of descent iterations versus the number of CG nodes. The top and bottom figures depict the results for CGs with and without absolute constraints, respectively. The lines in each figure show the average number of descents.

of CG nodes but small cliques and many ambiguities.

The results shown in Figure 3.12 are dependent upon a small modification to the function `GETRANDOMNODE(V)` in Algorithm 3.14. The modified version uses a simple rejection sampling strategy to favour the selection of nodes of higher degree. This change has a marginal effect on the search with absolute constraints (i.e., the average number of descent iterations becomes $c = 0.7n$ rather than $c = 0.8n$). However, for the search without absolute constraints, the average case falls from linear ($c = 0.45n$) to constant ($c = 200$). Clearly, the modified node selection strategy directs the search away from unlikely nodes and towards the relevant larger cliques within the sparse graph.

Finally, the number of descent iterations do not actually convey the total complexity of the maximum clique search algorithm, as each descent includes a call to the function `MUTUALLYADJACENTNODES(n)` (see Algorithm 3.16). This function performs a set intersection operation, which has linear complexity with respect to the degree of the passed node parameter.

The empirical results for the total search complexity, therefore, are shown in Figure 3.13.

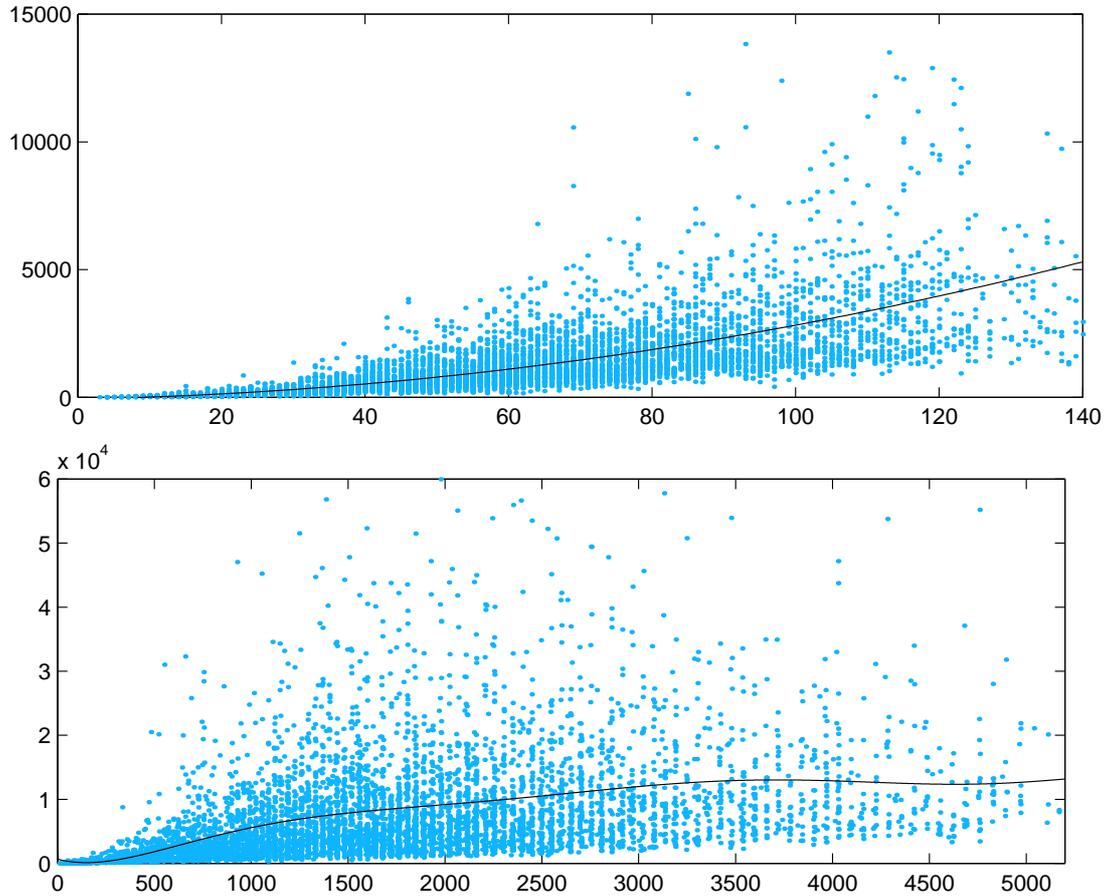


Figure 3.13: Maximum clique search complexity. These figures show the number of inner-loop iterations versus the number of CG nodes (with and without absolute constraints, respectively). The lines depict the average complexity.

The average case with absolute constraints is quadratic in relation to the number of nodes $c = 0.3n^2$. The average case complexity without absolute constraints is not so well defined. For the number of CG nodes between 0 and 700, the complexity appears quadratic $c = 0.01x^2$ but, after this point, it levels off. Whether this result is due to insufficient trials is unclear, but certainly the complexity for large n seems not worse than linear.

3.3.7 Ambiguity Management

The process of searching for the maximum clique finds many maximal cliques of equal or near maximum size. These cliques represent ambiguity in the CCDA result. There are two types of ambiguity that can arise (described here, for simplicity, in terms of just two maximal cliques). The first is due to environmental symmetry, where the association sets given by two cliques represent two completely different observer locations. One (or both) of these sets must be incorrect, and determining the correct clique is dependent on further information. The appropriate way to deal with this type of ambiguity is to use MHT, where

a different track is maintained for each clique. The second type of ambiguity is simple clutter, where both cliques represent essentially the same set of associations with some differing assignment-pairs due to sensor uncertainty and the close proximity of features.

In the case of multiple cliques, representing simple clutter, one of the following ambiguity management methods might be applied.

- Select the clique of maximum size that has maximum joint likelihood (i.e., joint nearest neighbour). Alternatively, choose the maximum joint likelihood clique from the set of cliques with size greater than a specified minimum (see Section 3.5.2 regarding minimum acceptable clique size).
- Retain only those associations common to the k largest cliques.
- Select the k largest cliques and perform JPDA with those associations that are ambiguous.
- Perform MHT with the k largest cliques (including, perhaps, a hypothesis that all association sets are false).

3.3.8 A Comparison of Optimal Assignment, JCBB and CCDA

This section compares the relative merits of optimal assignment (OA), JCBB and CCDA for finding a single most likely set of associations. They are compared according to two criteria: level of association constraint and computational complexity.

Constraint quality determines the number of associations that are rejected for a given statistical (i.e., NIS) probability of acceptance. The more constraint information available, the more associations become invalid. OA represents the best of the traditional multiple-target tracking data association methods (i.e., the best maximum likelihood method, this does not include JPDA or MHT).¹⁴ It incorporates injective mapping constraints and absolute constraints, and searches for the association set of maximum size with maximum summed log-likelihood.

The batch association methods, JCBB and CCDA, both incorporate injective mapping, absolute and relative constraints. Through the addition of relative constraints, these methods will always give a better (i.e., less ambiguous) result than OA for the same NIS threshold. Both methods search for the association set of maximum size with maximum joint likelihood, but their approaches to applying constraints are different. The JCBB algorithm selects an association set hypothesis and applies the absolute and relative constraints concurrently via calculation of the joint NIS. CCDA, on the other hand, applies these constraints explicitly and separately between feature pairs; this has a significant practical advantage as it enables the production of sensible results without absolute constraints.

An experimental comparison between OA and batch association is not given in this thesis. However, a comparison of JCBB and a greedy nearest neighbour association similar to OA is provided in [106].

¹⁴Arguably, the JAM method improves on OA by performing batch likelihood adjustment. However, the JAM has yet to gain popular acceptance and the experimental verification needed to make a proper comparison with OA.

The computational complexity of OA has a theoretical worst-case bound of $O(n^4)$ for n measurements and n targets [35], although the average-case complexity is reported to be considerably better. The implementation in [35] uses the ‘‘Hungarian’’ bipartite graph matching algorithm [84], and lower complexity might be possible using more recent bipartite matching methods (e.g., see [59, 14]).

JCBB performs incremental construction and search of an interpretation tree of joint association hypotheses. The joint NIS gate determines acceptable hypotheses and performs branch-and-bound pruning of the search space. The discussion in [106] does not provide a theoretical bound, but gives an empirical complexity estimate of $O(1.53^n)$, where n is the number of observed features (for fixed vehicle pose uncertainty).

The CCDA algorithm is composed of two parts: CG construction and maximum clique search. The first part takes $O(n^2 \log n^2 + km^2)$ time if the vehicle pose is unknown, where m and n are the number of measurements and targets, respectively, and k is found empirically to be $O(n)$. If the vehicle pose is known, CG construction may be performed by one of two methods. For the first, the same complexity $O(n^2 \log n^2 + km^2)$ applies but, for the second method, the complexity becomes $O(mn + |V|^2)$, where $|V|$ is the number of nodes in the CG. If the pose uncertainty is reasonably small, $|V|$ is $O(\min\{m, n\})$.

The maximum clique search complexity was found to be quadratic with the number of CG nodes (on average) if the vehicle pose is known, and not worse than linear if the vehicle pose is unknown (although more analysis is required for this second result). In these two cases the relationship between the number of nodes and number of features is $O(\min\{m, n\})$ and $O(mn)$, respectively. Therefore, assuming $m \approx n$, the average complexity in both cases, in terms of the number of observed features, is expected to be $O(m^2)$.

3.4 Application: Sensor-based Dead Reckoning

This application associates features between sequential laser scans to determine the change in pose of the observer. The experimental results have a two-fold objective: first, to demonstrate the CCDA algorithm in outdoor high-clutter environments and, second, to present a sensor-based alternative to odometry that is both more accurate and more reliable.

The two environments in which these tests are carried out are the parkland and internal road locations described in Appendix A. Both environments provide a challenging data association problem, particularly the latter with feature density ranging from extreme clutter to extreme sparsity.

3.4.1 Feature Extraction

The predominant landmarks in both environments are trees, and so feature extraction is governed by a point feature model for detecting tree trunks.

To begin with, the range-bearing measurements for a given laser scan are broken into clusters using Algorithm 3.17. Here, $\mathbf{z} = \{(r_1, \theta_1), \dots, (r_n, \theta_n)\}$ is the angle-ordered set of laser measurements, and the returned set $C = \{C_1, \dots, C_k\}$ is the resulting clusters. Clustering is performed on the basis of discontinuity between adjacent range measurements,

Algorithm 3.17: CLUSTER(\mathbf{z})

```

 $k \leftarrow 1$ 
 $C_k \leftarrow (r_1, \theta_1)$ 
for  $i \leftarrow 2$  to  $n$ 
   $\Delta_{\max} \leftarrow A + B \cdot \min\{r_i, r_{i-1}\}$ 
   $\Delta \leftarrow \text{abs}(r_i - r_{i-1})$ 
  if  $\Delta < \Delta_{\max}$ 
  do  $C_k \leftarrow C_k \cup (r_i, \theta_i)$ 
  else  $\begin{cases} k \leftarrow k + 1 \\ C_k \leftarrow (r_i, \theta_i) \end{cases}$ 
return ( $C$ )

```

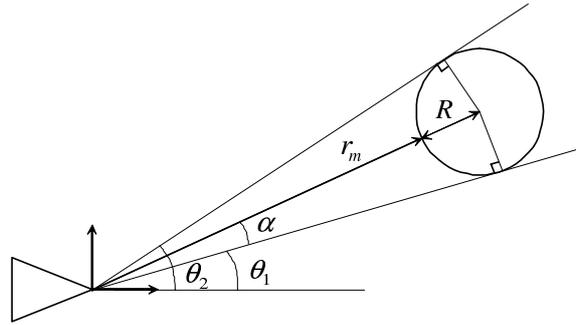


Figure 3.14: Point feature model. Given a foreground cluster (i.e., a cluster with both edge points closer than the adjacent clusters), the parameters of a circle can be calculated from the minimum range measurement r_m and the angles of the two edge measurements.

and is tuned by adjusting the gains A and B . In these experiments, the constant gain $A = 0.07m$ and the proportional gain $B = 0.04$.

Clusters are selected from the ordered set C if they are in foreground. That is, a cluster i is selected if its first and last range measurements are shorter than the last and first range measurements of clusters $i - 1$ and $i + 1$, respectively. These foreground clusters define point features according to the model shown in Figure 3.14, which expresses the range-bearing of a circle origin in terms of the cluster's bounding angles $\{\theta_1, \theta_2\}$ and minimum range r_m .¹⁵ From this diagram, it can be seen that

$$\sin \alpha = \frac{R}{r_m + R} \quad (3.27)$$

¹⁵An alternative circle extraction model is given in [67], which estimates the circle parameters from the entire set of cluster measurements using an EKF. However, empirical laser tests indicate that range values at sharp discontinuities tend to be very unreliable, and this is the rationale for the simple model presented in this thesis—which uses only angle information at the cluster edges.

where $\alpha = \frac{\theta_2 - \theta_1}{2}$. Rearranging Equation 3.27, the circle radius is given by

$$R = \frac{r_m \sin \alpha}{1 - \sin \alpha} \quad (3.28)$$

Therefore, the range and bearing measurement $\mathbf{z}_p = [r_p, \theta_p]^T$ is

$$r_p = r_m + R = \frac{r_m}{1 - \sin \alpha} \quad (3.29)$$

$$\theta_p = \frac{\theta_1 + \theta_2}{2} \quad (3.30)$$

The individual laser measurements are assumed to be independent, with Gaussian uncertainty in range and bearing, such that each measurement $\mathbf{z}_i = [r_i, \theta_i]^T$ has covariance

$$\mathbf{P}_i = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$$

Therefore, if the circle origin is defined as a function of the parameters $(r_m, \theta_1, \theta_2)$

$$\begin{bmatrix} r_p \\ \theta_p \end{bmatrix} = \mathbf{h} \left(\begin{bmatrix} r_m \\ \theta_1 \\ \theta_2 \end{bmatrix} \right) \quad (3.31)$$

so that the Jacobian $\nabla \mathbf{h}$ is given by

$$\nabla \mathbf{h} = \begin{bmatrix} \frac{\partial r_p}{\partial r_m} & \frac{\partial r_p}{\partial \theta_1} & \frac{\partial r_p}{\partial \theta_2} \\ \frac{\partial \theta_p}{\partial r_m} & \frac{\partial \theta_p}{\partial \theta_1} & \frac{\partial \theta_p}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} \frac{1}{1 - \sin \alpha} & -\frac{r_m \cos \alpha}{2(1 - \sin \alpha)^2} & \frac{r_m \cos \alpha}{2(1 - \sin \alpha)^2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (3.32)$$

then the foreground point covariance \mathbf{P}_p can be approximated by linearised transformation of the measurement uncertainties.

$$\mathbf{P}_p = \nabla \mathbf{h} \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \nabla \mathbf{h}^T = \begin{bmatrix} \frac{1}{(1 - \sin \alpha)^2} \sigma_r^2 + \frac{r_m^2 \cos^2 \alpha}{2(1 - \sin \alpha)^4} \sigma_\theta^2 & 0 \\ 0 & \frac{1}{2} \sigma_\theta^2 \end{bmatrix} \quad (3.33)$$

Notice that this model produces a correct estimate of the point range-bearing covariance only if the object observed is, in fact, circular, and it assumes that the minimum range measurement r_m is actually an observation of this circle's closest point. However, clusters are classified as points based on the sole criterion that they are foreground, which means that non-circular clusters can be selected, and are often poorly represented by this model (e.g., see Figure 3.15). To cater for non-ideal feature classification, it is necessary to inflate the observation uncertainty; also, to limit the possibility of gross misclassifications, a maximum radius estimate is applied (i.e., only keep those points satisfying $R < R_{MAX}$). Typically, the tree trunks in the two experimental environments are well approximated by the circular model as depicted in Figure 3.16.

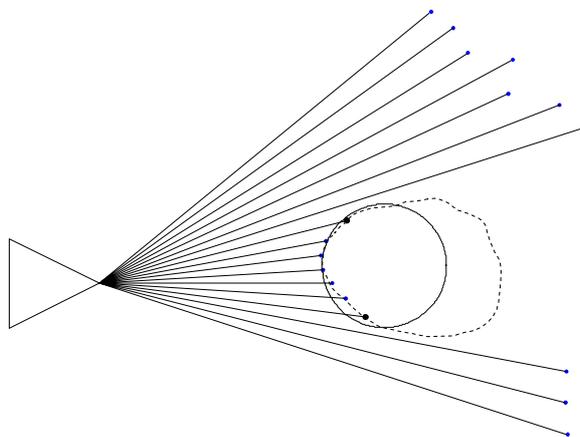


Figure 3.15: Feature misclassification. A non-circular object is poorly represented by the circular foreground point model.

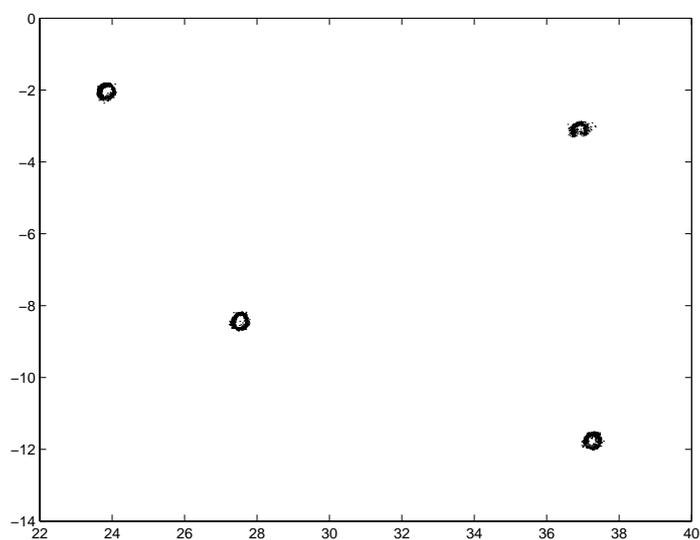


Figure 3.16: Tree trunk observations in the park environment. Using SLAM pose estimates, the unprocessed laser points are recorded over a series of passes to illustrate the circular shape of these four trunks.

3.4.2 Dead Reckoning Implementation Details

The steps of this dead reckoning algorithm follow from the CCDA algorithm in Section 3.3, and the relative pose estimation method in Appendix C.3. Given a set of change-in-pose estimates $\{\hat{\mathbf{x}}_\delta, \mathbf{P}_\delta\}$ between sequential scans, the accumulated dead reckoning estimate is found using the equations in Example C.1. Therefore, to avoid unnecessary repetition, the discussion below just provides additional details regarding (i) calculation of the change-in-pose \mathbf{x}_δ , and (ii) application of the CCDA algorithm.

The initial estimate for a particular \mathbf{x}_δ is based on the prior information afforded by the change-in-pose estimate of the previous scan-pair. Thus, the relative pose calculations in Appendix C.3 use Equations C.17 and C.18 rather than Equations C.15 and C.16. The prior estimate is given by a simple constant-velocity inertial model as follows.

$$\hat{\mathbf{x}}_{prior} = \hat{\mathbf{x}}_\delta \quad (3.34)$$

$$\mathbf{P}_{prior} = \mathbf{P}_\delta + \mathbf{P}_{acc} \quad (3.35)$$

where $\hat{\mathbf{x}}_\delta$ and \mathbf{P}_δ are the previous change-in-pose estimate, and \mathbf{P}_{acc} represents the increase in uncertainty due to possible accelerations over the time interval. Note, the prediction of constant $\hat{\mathbf{x}}_\delta$ is possible since the laser information arrives after equal time intervals (4.7 Hz in these trials). The value of \mathbf{P}_{acc} is determined by a model of the maximum possible vehicle accelerations; in these trials, \mathbf{P}_{acc} was fixed at

$$\mathbf{P}_{acc} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.03 \end{bmatrix}$$

Batch data association was performed using the first of the tracking CCDA methods (see Section 3.3.4). This implementation used a basic form of the maximum clique algorithm, which returned, as a “best” solution, the largest association set found (or, if multiple equal largest sets exist, the first largest set found). Significantly better results could probably be obtained if other association sets were also considered, but these results were sufficient to demonstrate the utility of the CCDA algorithm.

The test environments displayed a high degree of variability in landmark density at different locations, resulting in batch associations ranging from less than 3 in some regions to as great as 70 in others. For each scan-pair, a minimum batch size was applied to determine whether the association set was deemed reliable and, while more sophisticated measures may be possible (considering factors such as local feature density), this simple threshold gave reasonable results (see Section 3.5.2 below for further discussion). If, for a given scan-pair, insufficient associations are obtained, then the estimated change-in-pose is equal to the prior estimate, and this in turn becomes the prior for the next scan-pair (with expanded uncertainty $\mathbf{P}_{prior} = \mathbf{P}_\delta + \mathbf{P}_{acc}$). There is a basic tradeoff, therefore, between association reliability and the possibility of prolonged periods of tracking failure, and batch-size thresholds of between 3 and 7 were found to work best in these trials.

3.4.3 Results

Wheel-encoder odometric data was logged during both the trials presented below, and an odometry-based dead reckoning estimate calculated using the vehicle model parameters

that were used in [67, 65] (for the same vehicle). These model parameters were tuned to give optimal results, and the odometric results shown here are assumed to be of the best possible quality. The laser result, on the other hand, required no detailed kinematic model and operated with just a coarse dynamic process model and a simple observation model.

The dead reckoning results for the park environment are shown in Figure 3.17, with the top figure based on wheel-encoder information and the bottom on laser scans. The superior accuracy of laser-based dead reckoning is readily apparent, particularly in heading. In general, odometry tends to provide a smoother estimate, but is subject to biases (many of which are not easily modelled, such as slip during turning). Laser-based dead reckoning, while noisier, does not possess such significant bias.

The internal road result, shown in Figure 3.18, exhibits the same characteristics over a longer distance (approximately 6 km). Several regions of this environment produced relatively inaccurate laser-based estimates due to lack of observable features, but overall the trajectory retains the geometry of the roadway. The encoder-based estimate, on the other hand, rapidly diverges into incoherence.

3.5 Remarks

The two matters discussed in this section are concerned with (i) the usefulness of geometric features in general unstructured environments, and (ii) determining a minimum association set threshold for CCDA.

3.5.1 Utility of Simple Geometric Features

Simple geometric primitives, such as points and lines, are often criticised¹⁶ as insufficient for general environments and a current trend [93, 118, 132, 95, 48] is to try and develop more general representations (e.g., scan correlation, see Chapter 4). A typical rationale for rejecting simple features is the problem of misclassification; for example, if a smoothly curved surface is represented by a set of line segments, these lines may shift arbitrarily. Also, methods like scan correlation use all the available observation information, and not just the parts that can be classified.

It is argued here, however, that geometric models are probably more universally appropriate than they are given credit. Point locations such as edges (i.e., range discontinuities) and corners occur in a great many environments. Batch association, and the feature management methods presented in Chapter 5, can be used to reject unstable features without harm. Another important factor to note is that data association imparts significant information and feature-based methods may produce *better* results than scan correlation.

Some environments, such as mining tunnels and subsea, do not yield stable geometric features and require more general representations. However, this does not preclude the use of the EKF stochastic SLAM framework. Fundamentally, there is confusion between the use of geometric (point) feature models and the estimation of the point location of an object. The former may well not apply, but the latter is generally applicable to any static

¹⁶For example, geometric feature-based methods have been described as “less accurate and less robust” than scan matching for laser-based localisation in an indoor environment [93].

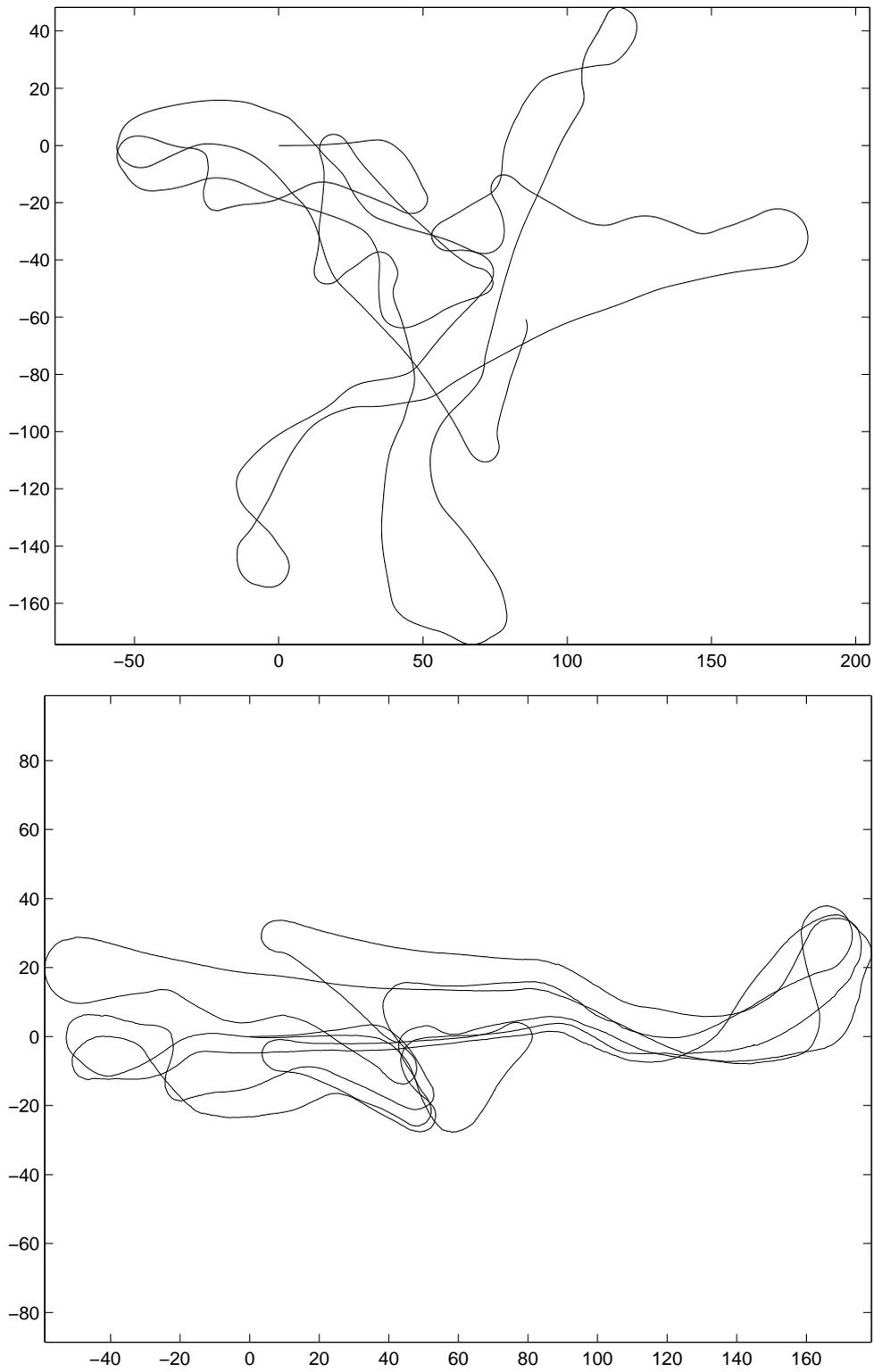


Figure 3.17: Encoder-based (top) versus laser-based dead reckoning in the park environment.

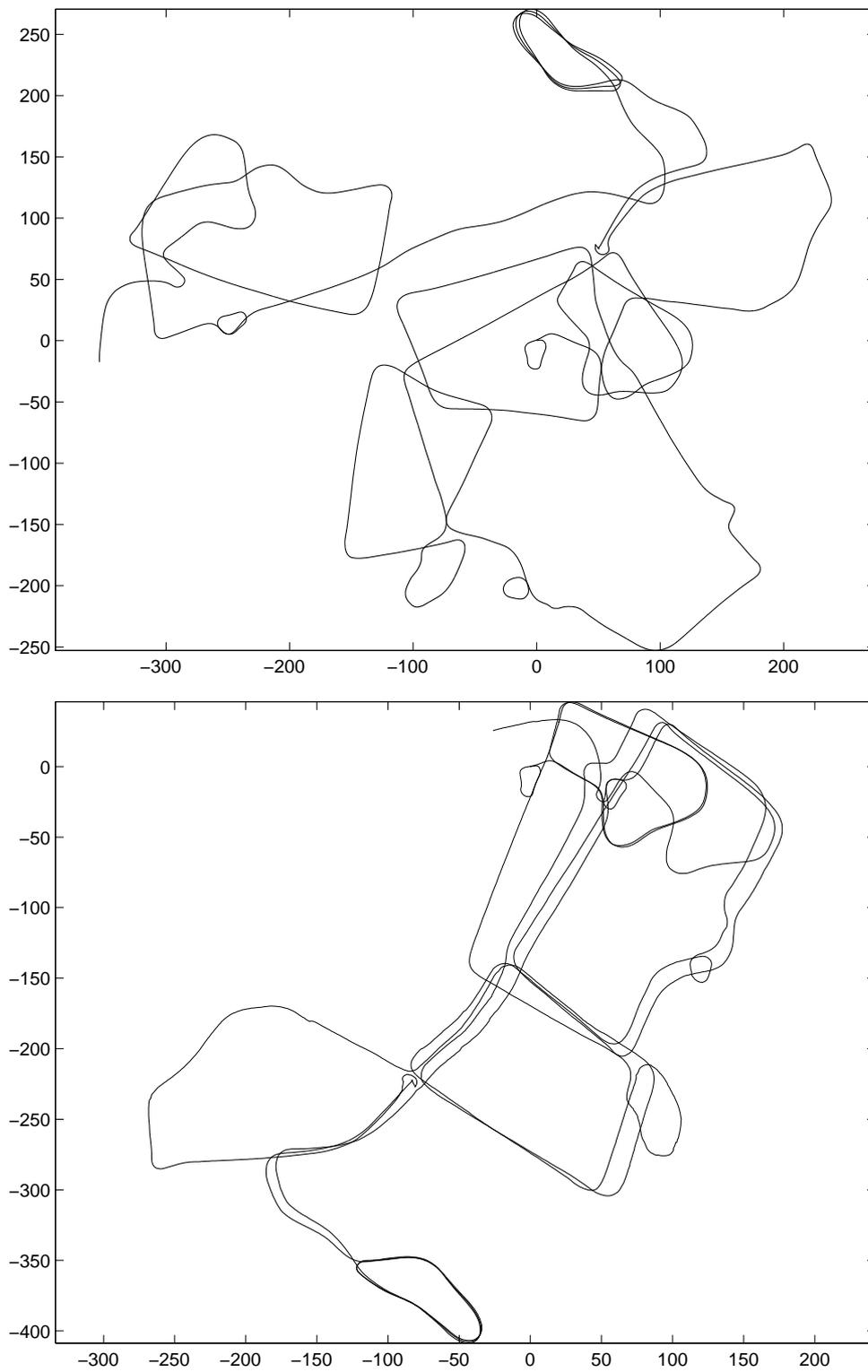


Figure 3.18: Encoder-based (top) versus laser-based dead reckoning in the internal road environment.

map. For example, an object might be represented by a set of unprocessed measurement data, which serves as a static model of the object. Subsequent observations of the object can then use scan correlation to estimate the object location relative to the vehicle (i.e., as a relative pose (x, y, ϕ) and uncertainty). By approximating the relative pose uncertainty with a Gaussian, the global object pose can be maintained in the EKF SLAM state vector. This approach is suggested as a future research direction combining scan correlation and SLAM in Section 7.2.2. Essentially, an object might not possess a simple parametric feature model but still defines a point pose in space (its local coordinate axis) which is static and approximately Gaussian.

3.5.2 Minimum Acceptable Batch Size

Batch data association, as described in this chapter, produces sets of associations that are mutually compatible given their combined constraints. From this it is possible to obtain a “best” association set consisting of the greatest number of associations. However, selecting a best set, or set of best sets, is not sufficient for ensuring *correct* association (i.e., selecting the best of a bad batch is not good enough). For the SLAM problem, the real requirement is a measure of association reliability; it is better to reject all associations, or perform MHT where one hypothesis rejects association, than to risk accepting an incorrect association set.

A likelihood of association set “correctness” is a function of many factors: (i) the number of associations in the set, (ii) the quality of the individual constraints and correlations, (iii) the geometric distribution of features, (iv) the observer pose uncertainty, (v) the presence of clutter, occlusions, and dynamic objects, (vi) and so on. In this thesis, association reliability is not properly quantified and defining an appropriate measure remains an open question. The approach used in the experiments of this chapter considers only property (i) above, such that a threshold on minimum batch size is used as a blanket criterion. This is a very suboptimal approach; a low threshold is fragile in dense regions due to symmetries and a high threshold is fragile in sparse regions because of possible tracking loss.

3.6 Summary

This chapter addresses the data association problem for localisation and SLAM. First, it reviews the data association methods developed for target tracking, where the basic test for accepting or rejecting an association is the NIS (or Mahalanobis distance) validation gate. For tracking a single target in clutter, there are three main procedures for resolving association ambiguity: nearest neighbour, PDA, and MHT. These methods have analogues for tracking multiple targets in clutter: optimal assignment, JPDA, and MHT.

If a set of targets, or a set of observations, are correlated within themselves, then more robust data association can be achieved if they are processed as a batch. Two batch association methods are reviewed: JCBB and MCS data association. MCS data association is unique in finding associations without prior knowledge of the observer pose.

The CCDA algorithm is presented as extension to the MCS data association method. This algorithm is based on a “correspondence graph” (CG) which is constructed from the relative and absolute constraints between feature pairs. The maximum clique of the CG

defines the largest set of mutually compatible associations. Every other clique in the CG represents an alternative (ambiguous) association set hypothesis.

The CCDA method is demonstrated with an implementation of laser-based dead reckoning in two outdoor environments. These estimates are shown to be greatly more accurate than encoder-based odometry.

Chapter 4

An Alternative to Feature-based Data Association: Scan Correlation

Feature-based data association is a viable solution in environments where static landmarks exist suitable for classification as geometric primitives. However, in some environments (e.g., see Figure 4.1), it may be difficult to define appropriate parametric feature models, and more reliable association is possible through direct correlation of unprocessed observation information with the map. Direct correlation is also a useful alternative in environments amenable to feature-based methods, as it incorporates all available sensor information (i.e., does not reject unmodelled data) and avoids imperfect model-based classification.

Unprocessed data correlation, also called scan correlation or range-image registration, is the process of aligning an observed set of (2-D or 3-D) points with a reference point set. Thus, scan correlation may be defined as a function of the relative pose between the two data sets.

For the mobile robot localisation problem, scan correlation is useful as a mechanism to facilitate sensor-based dead reckoning, localisation from an *a priori* map [38], and off-line map generation [128, 22, 82]. Presently, the implementation of real-time stochastic SLAM using scan correlation is considered intractable, but recent proposals regarding incremental EM-based SLAM [130] and hybrid topological-metric SLAM [92, 68] show promise for tractable and consistent correlation-based solutions.

A large number of correlation techniques have been proposed in the literature, as reviewed in the next section. However, most are based on heuristic cost functions and lack a theoretical probabilistic derivation. The motivation of this chapter is to present a theoretically grounded approach to scan correlation—derived from a stochastic sensor model—so that the correlation procedure provides an accurate estimate of the relative pose uncertainty.

This chapter presents a method and justification for performing scan correlation, with discussion of the following topics.

- Current methods for performing scan correlation are reviewed, with particular attention to their deficiencies concerning sensor modelling and uncertainty estimation.
- For the probabilistic representation of scan measurements, several different parameter-spaces are considered. In particular, given a scan of 2-D points, the merits and

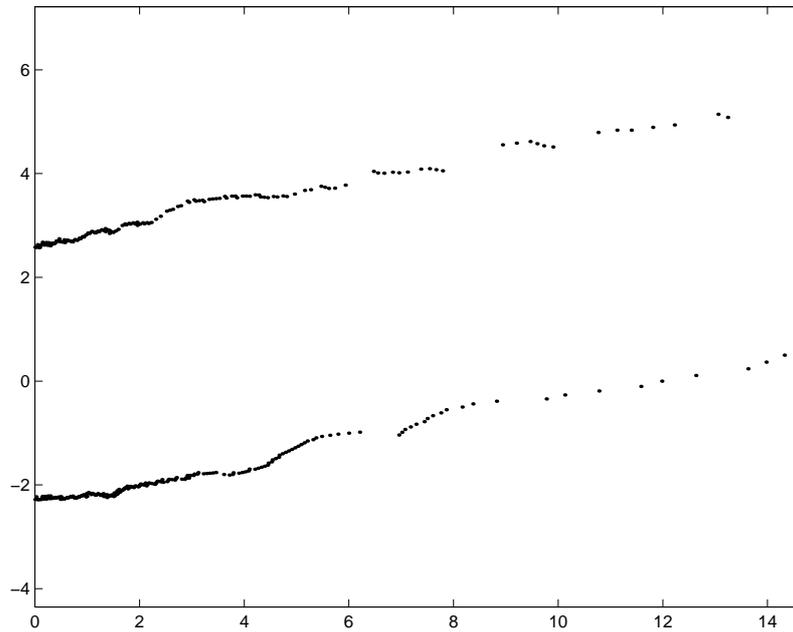


Figure 4.1: Unprocessed laser scan of a mining tunnel. In this environment it is difficult to extract stable parametric features but there exists ample information for scan correlation given the surface texture of the tunnel walls.

limitations of a 2-D Cartesian-space representation are examined and compared with those of higher-dimensional parameter-spaces.

- The 2-D parameter-space causes a scan to be interpreted according to a point-target model. Assuming this model, a Bayesian likelihood function for scan correlation is derived.
- Conversion from a scan of unprocessed point data to a 2-D probability distribution is implemented using a *sum of Gaussians* representation. This representation is shown to facilitate efficient calculation of the scan correlation likelihood function.
- Scan correlation is applied to unprocessed laser data (represented as Gaussian sums) to perform two practical localisation tasks: maximum likelihood dead reckoning and particle filter localisation.

Caveat. This chapter is quite disparate from the rest of the thesis, which is primarily concerned with feature-based SLAM. The focus of this chapter is not SLAM but simply the alignment of two point data scans in environments not suited to feature extraction. In this chapter, scan correlation is investigated in a probabilistic manner, and the reader is assumed to be familiar with the basic concepts of recursive Bayesian estimation. In particular, it is necessary to understand terms such as *probability density function* (PDF) and *likelihood function*. For a brief introduction to these concepts, refer to Appendix D.

```

Algorithm 4.1: ICP( $P_o, P_r, \mathbf{x}_0$ )

 $\mathbf{x} \leftarrow \mathbf{x}_0$ 
while not Converged
  do
     $E \leftarrow \emptyset$ 
    for each  $p \in P_o$ 
      do
         $p' \leftarrow \text{TRANSFORMPOINT}(p, \mathbf{x})$ 
         $q \leftarrow \text{FINDNEARESTNEIGHBOUR}(P_r, p')$ 
         $E \leftarrow E \cup \{p, q\}$ 
     $\mathbf{x} \leftarrow \text{CALCULATELEASTSQUARESPOSE}(E)$ 
return ( $\mathbf{x}$ )

```

4.1 A Review of Point Data Correlation Techniques

This section examines a selection of commonly used scan correlation methods, with a particular focus on whether these methods incorporate appropriate sensor uncertainty models. The following review presents two non-probabilistic techniques, *iterative closest point* (ICP) and the *angle histogram*, and several other methods that may be implemented in a probabilistic fashion including grid correlation and particle filter localisation.

In the discussion below, the two scans are termed the reference scan and the observation scan, where the fixed reference scan defines the base coordinate frame. Correlation, therefore, involves finding the pose of the observation scan relative to this base coordinate frame.

4.1.1 Iterative Closest Point

ICP [13, 143] is arguably the most commonly used range-image registration technique, with its popularity due mainly to its simplicity and efficiency. The basic algorithm is shown in Algorithm 4.1. Let $P_o = \{p_1, \dots, p_m\}$ represent the observation point set and $P_r = \{p_1, \dots, p_n\}$ be the reference point set. The relative pose of the observation set is denoted \mathbf{x} . The algorithm is initialised with an initial pose guess \mathbf{x}_0 and, until the estimated pose satisfies some convergence criterion, it is iteratively refined by a process of point-to-point data association and least-squares transformation. Each point $p \in P_o$ is first transformed to the reference coordinate frame using the current pose estimate, and then associated to its nearest neighbour in P_r . The original point p and its associate q are added to an association set E . Finally, the pairs in E are used to calculate the relative pose that minimises the least-mean-squared error between the associated points.

For each iteration, the nearest neighbour search can be performed in $O(m \log n)$ time using an efficient search data structure (e.g., k-d trees [12, 101]), and the least-squares calculation has a closed form solution [2, 73, 93] that can be performed in $O(m)$ computations. Convergence of the algorithm occurs when the nearest neighbour for each $p \in P_o$ does not change between iterations. However, it might also be determined by a least-mean-squared residual threshold, or simply a fixed number of iterations k —in which case the total

algorithm complexity is $O(km \log n)$.

There are three basic shortcomings to the ICP algorithm. First, it performs explicit point-to-point data association each iteration (n.b., without injective mapping constraints), which introduces error since the points in each scan represent a surface and not a set of discrete locations.¹ Second, ICP converges to a local minima and so requires a good initial pose guess to find the global minimum. And third, the ICP result is a least-mean-squares estimate where each association pair is equally weighted; there is no direct means to incorporate modelled sensor uncertainty or to obtain an estimate of pose uncertainty in the solution.

A precursor to ICP [33], which uses a reference map of line segments and performs correlation with an observation point set, is worth noting for its efficiency and use in current localisation systems [69, 68]. Also, a large selection of ICP variants have been proposed in the literature over the past decade. These include improved convergence rate [93, 115], models for point location error [43], and a method for estimating the final pose uncertainty [124] (although neither of the latter two variants are derived from models of sensor uncertainty).

4.1.2 Angle Histogram

The angle histogram method [71, 139] of scan correlation is specific to a planar sensor with scanning-laser qualities. That is, the sensor must return a set of range-bearing measurements equispaced in bearing. In addition, each scan is expected to cover a full 360° so as to provide a circular (closed) sequence of measurements.

The essence of the angle histogram algorithm is as follows. Each measurement (r_i, θ_i) in a scan is projected into observer-centred Cartesian space. Given two consecutive measurements, $\mathbf{v}_i = (x_i, y_i)$ and $\mathbf{v}_{i+1} = (x_{i+1}, y_{i+1})$, the angle of their vector difference is given by

$$\alpha_i = \arctan \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

The angle histogram is generated from the set of angles for each pair of adjacent scan measurements. The shape of this histogram is invariant to pure rotation, which appears as a phase shift. Thus, the amount of sensor rotation may be found by cross-correlation of the two histograms. The relationship between phase shift and sensor rotation is valid even when combined with a small quantity of sensor translation. Therefore, the change in sensor pose is found by first rotating the observation scan by the calculated angle offset, and then finding the translation offset using x and y histograms.

The angle histogram method does not appear to offer any advantage over ICP. It is neither more efficient, more accurate, or simpler to implement. Perhaps the one benefit of the angle histogram is that it does not require explicit point-to-point associations.

¹The error due to discrete associations is usually small if the associated points represent nearby locations. However, outlier associations, where associated pairs are very distant, can skew the least-squares solution. A simple measure to improve outlier robustness is to ignore pairs with distance greater than a given threshold.

4.1.3 Occupancy Grid Correlation

Occupancy grid maps (described in Section 2.1.1) use scan correlation to perform registration of the observed scan with the map [50, 128, 118, 82]. The method of registration is 2-D cross-correlation, which is usually implemented in the same manner as template matching in image processing. That is, the observation grid is shifted by a given pose, and the correlation value for this pose is calculated by (i) multiplying the overlapping grid cells together, and (ii) summing this result for each cell in the observation grid. (Note, since the observer pose varies in x , y , and ϕ , the search space is 3-dimensional. Thus, the two grids do not overlap cleanly and interpolation techniques are required, as shown in [128] for example. Also, the search-space is much greater than for 2-D, and cross-correlation may be computationally infeasible without *a priori* bounds. This is also dependent on the tradeoff between efficiency and granularity as discussed in Section 2.1.1.)

Occupancy grids represent sensor information in a high-dimensional parameter-space, where each cell has an separate variable for its state between 0 and 1. The value and correlation of these states is updated in a non-Bayesian manner according to geometrically derived sensor models (e.g., see the sonar model in [50]). That is, a sensor is modelled spatially, in terms of probability of occupancy, rather than by directly relating the measurement uncertainty to the state of each cell parameter. This means that the relationship between the sensor uncertainty and the uncertainty obtained from cross-correlation is not precisely defined. Cross-correlation may still be probabilistic, but is not associated to the sensor model in a Bayesian sense.

While occupancy grids may achieve good scan correlation results by virtue of the large quantity of information stored in their grid cells, they fail to exploit the higher accuracy and robustness available from a Bayesian model, and do not provide a theoretical justification for the resulting relative pose uncertainty.

4.1.4 Probabilistic Methods

A number of recent scan correlation techniques claim to operate according to probabilistic principles, meaning that correlation is driven by a likelihood function derived from a stochastic observation model. Usually these methods are presented as *maximum likelihood* algorithms, which attempt to find the relative pose value that maximises the likelihood function, but a few proposals demonstrate full probability density estimation, combining the likelihood function with prior information to calculate an *a posteriori* PDF of the pose.

Two forms of maximum likelihood correlation appear commonly in the literature. The first is ICP-like, iteratively searching for nearest neighbour data associations for each point in the observation scan [108, 110]. Each association is given a likelihood value, and the product of these likelihoods defines the scan likelihood. A maximum likelihood pose can be found by sampling pose values within the solution space, usually employing a greedy optimisation algorithm to converge to a local maximum. The problem with these ICP-based methods is that the likelihood models do not accurately represent the sensor uncertainty. They are typically based on geometric properties rather than a proper sensor model (e.g., in [108] the likelihoods are simply functions of nearest-neighbour distance) and, more importantly, they rely on an incorrect model of explicit point-wise association. The algorithm presented

in [110] incorporates measures to account for correspondence error, and is a promising ICP variant, but it remains an *ad hoc* geometric fix for these fictitious discrete associations.

The second common maximum likelihood correlation technique is grid-based, typically using occupancy grids [50, 22, 82]. However, an alternative grid-based concept is probability grids (see [125], for example), which are fundamentally different to occupancy grids. While a 2-D occupancy grid defines a high-dimensional parameter-space equal to the number of its grid cells, a 2-D probability grid simply represents a 2-D parameter-space. (In [96, 95], a probabilistic grid is used to represent the uncertainty distributions of unprocessed sonar ping returns. Note the difference between the sonar model in [96, 95] and the occupancy grid sonar model in [50].) The main weakness of grids as a probability representation is that they artificially discretise the distribution (i.e. pixelation), which involves a considerable loss of information. In some cases, better PDF representation might be obtained using, for example, sums of Gaussians, wavelets, NURBS, and so forth. Using a 2-D probability representation for scan correlation (of 2-D scans) is a primary focus of this chapter.

Estimation of the full *a posteriori* PDF is rarely applied to the scan correlation problem because of computational complexity. However, a recent advance in general non-linear estimation called particle filtering (see Appendix D), which approximates general PDFs with samples, enables tractable computation of many problems previously considered infeasible. In [132], a particle filter is used to perform localisation from an *a priori* map via scan correlation.² This implementation permits direct estimation of multi-modal pose distributions and so avoids the fragility inherent in the maximum likelihood approach.

4.2 A Probabilistic Representation for Unprocessed Data

To perform probabilistic scan correlation, it is necessary, first, to define an appropriate parameter-space for the scan data and, second, to derive a Bayesian likelihood function for the relative pose between two scans. This section addresses the first requirement: the definition of a parameter-space, and the conversion of a scan of real data to a joint PDF in this space. The derivation of a likelihood function, having specified a parameter-space, is presented in Section 4.3.

For clarity, this discussion is given in terms of a 2-D laser scan consisting of n range-bearing measurements. Each laser measurement observes the presence of a (diffuse reflective) surface within the envelope of its beam-width and maximum range. The actual location of the observed surface is uncertain and is modelled by a Gaussian PDF. Thus, for a set of n measurements, there is a set of n Gaussians each representing a discrete surface location. These surface points are distinct from each other but may be correlated due to their proximity, the smoothness of the environment, and the beam-width of the sensor.

The objective of this section is to represent the scan of measurements so that scan correlation is both accurate and consistent. Two basic forms of state-space parameterisation are considered. The first is a projection from 2-D measurement space to a 2-D Cartesian space,

²The implementation in [132] presents an interesting simplification to reduce computational load. The likelihood function is built on a ray-tracing model, which makes an implicit data association for each range-bearing measurement with its map intersection. Essentially, this implies that uncertainty is present in the range measurement only (modelled as Gaussian) and both the bearing measurement and the map are perfectly known.

which represents world-coordinates. The second is higher-order, where the measurements are mapped to a non-Cartesian parameter space of dimension greater than two.

The following discussion examines the transformation of the individual measurement uncertainties to these spaces, and considers whether each parameter-space is a reasonable representation of the original scan. The utility and limitations of each representation for scan correlation is considered, with particular attention to the criteria below.

- **Generality.** The transformation from measurement to parameter-space should not require restrictive models, as this is the primary reason for choosing scan correlation over feature-based methods.
- **Data association.** Ideally, scan correlation should avoid data association thresholds.
- **Accuracy and consistency.** The resultant pose uncertainty from scan correlation should be minimal, and consistent in the (weak) sense that the likelihood for the true pose is non-zero.
- **Tractability.** Scan correlation should be computationally feasible in real-time.

4.2.1 Representation as a Two-Dimensional PDF

A 2-D PDF is generated from a scan of measurements by transforming each range-bearing Gaussian to sensor-centred Cartesian space, and accumulating these distributions by scaling and summing. The resulting Gaussian sum³ defines an uncertainty distribution over the 2-D Cartesian parameter-space. The details for this operation are presented in Section 4.4.

The 2-D parameter-space implies that the scan represents a *single point target*. This is implicit in its dimensionality; the PDF defines the uncertainty of a random vector (x, y) for which there can exist only a single true value (x_t, y_t) .

This point target model has several advantages. First, it is general; conversion from measurement uncertainties to the Cartesian PDF does not involve any feature models. Second, since the model implies a single entity, there is no data association problem for scan correlation. Alignment is determined by cross-correlation of the two scan PDFs (see Section 4.3 below). Third, cross-correlation of two Gaussian sums can be computed efficiently (see Section 4.4) and does not incur the artificial pixelation of grid methods. And finally, this model appears to produce accurate results in practice (see Sections 4.5 and 4.6).

Nevertheless, the 2-D parameter-space representation has two significant difficulties. The first is obvious; the point target model is not a true representation of the actual data. Plainly, the measurements describe n discrete points of an arbitrarily complex surface, not the location of a single point target. The second problem is more subtle. The individual measurement Gaussians are scaled before being added to the Gaussian sum, and the values of these scaling terms, necessary for good results in practice, are difficult to justify in theory (this issue is addressed further in Section 4.4.2).

³The transformed Gaussians will no longer be Gaussian due to the non-linear polar-to-Cartesian function. However, Gaussian approximations to these PDFs are reasonable in practice. Also note, the PDFs for each measurement must be combined by summation rather than multiplication; for further explanation, see Section 4.7.

The 2-D space representation effectively converts the interpretation of a scan from a set of discrete surface measurements to a set of possible locations for a point target. The distribution of one Gaussian in the sum says “the point target could be here” while another says “or here” and so forth. Basically, the scan measurements can be interpreted as a 2-D analogue of a sonar ping, where each sampled amplitude of the return signal indicates the likelihood of a point target at a certain range.

While the 2-D model has serious imperfections, it is not entirely incorrect either. For instance, for a single measurement, both the true and contrived interpretations state that “there is a point target somewhere in this distribution.” More importantly, provided two scans share at least some commonality, scan correlation using the 2-D PDF representation will result in a consistent likelihood function for their relative pose. (That is, the likelihood of the true relative pose will be non-zero, provided the scans actually share common information and the individual measurement uncertainties are modelled consistently; they need not be modelled as Gaussian.) This property arises from the likelihood function being computed via cross-correlation of the two scans (see Section 4.3), which will give a non-zero result for the true pose even if there is only common information from a single measurement in each scan. Therefore, the 2-D model seems also to offer robustness to outliers and partial views; this is examined further in Section 4.4.6.

The 2-D parameter-space model satisfies the four essential criteria set out at the beginning of this section. However, in light of its deficiencies, a number of higher dimensional models are considered as to their ability to better represent the scan data and to facilitate scan correlation.

4.2.2 Higher Dimensional Alternatives

Since a 2-D parameter-space is insufficient to properly describe the scan, a complete solution necessitates a higher-order parameter-space. Three higher-dimensional possibilities are considered in this section: feature-based parameters, measurement parameters, and occupancy grids.

A feature-based parameter-space is obtained by specifying feature models, and clustering and classifying the sensor measurements according to these models; this is simply traditional feature-based tracking as covered in Chapter 3. Where applicable, the feature models transform the measurement data to feature-space $[\mathbf{f}_1, \dots, \mathbf{f}_k]^T$, and each new feature specifies an augmentation of the space. The drawbacks of this approach are well known. Representation of the measurement data in feature-space is subject to restrictive models, which ignore a high proportion of the available information and which may not be appropriate in certain environments. Also, performing scan alignment incurs the data association problem—finding correspondences between features from the two scans—which can be fragile.

A preferable approach is one that avoids both feature models and data association. One such approach is to define the parameter-space by the measurements themselves $[r_1, \theta_1, \dots, r_n, \theta_n]^T$. Thus, a scan is represented without models and data association is not meaningful, since a measurement from one scan observes a point on a continuous surface and will not directly correspond to any particular measurement in another scan. In this case, the full Bayesian solution to scan correlation is to define a likelihood function that

models the probability of obtaining the second scan, conditioned on the state of the first scan and their relative pose. However, definition of such a function is complicated because, for any given relative pose, the likelihood of a measurement in the second scan may be related to the existence of any number of measurements in the first.

Even if a reasonable likelihood function could be defined, the large state-space involved, consisting of the relative pose and the measurements of the first scan, presents a still greater problem. A recent general solution to the problem of tracking states without data association, called *unified tracking*, is presented in [125]. The implementation of this method is insightful particularly with regard to the tractability of tracking without data association. The authors state that unified tracking “is computationally infeasible for problems involving even moderate numbers of targets” [125, page 205]. They claim that it is practical for up to two targets, but that efficient approximations to the full solution are necessary for applications with greater target numbers. Therefore, it seems that, for higher-dimensional parameter-spaces, there is no current, tractable solution to scan correlation without some form of data association (which might be maximum likelihood, multiple hypothesis, heuristic, or otherwise).

Occupancy grids are a high-dimensional parameter-space that do not require feature models and permit feasible scan correlation. However, the occupancy grid solution is *ad hoc* in the sense that the uncertainty and correlations of the cells are updated in a non-Bayesian manner. This means that there is no theoretical relationship between the sensor uncertainty and the uncertainty of the scan correlation result. Also, data association between parameters is determined by an *ad hoc* metric of geometric proximity during cross-correlation. For the purpose of scan correlation, occupancy grids offer no advantage over the 2-D parameter-space model. They have no better theoretical justification, are expected to be less accurate (due to pixelation), and are likely to be less efficient (depending on their granularity).

Of the higher-dimensional representations, the first two fail to meet the criteria specified at the beginning of this section. The feature-based parameter-space uses restrictive models and data association, and the measurement-space representation is not tractable. Occupancy grids satisfy the criteria in principle (although they implement heuristic data association), but are a non-Bayesian solution that possess no better justification than the 2-D parameter-space approach.

4.2.3 Application of the 2-D Representation

In environments where reliable feature extraction is difficult, the 2-D parameter-space model seems to be a reasonable, if imperfect, representation and superior to its higher-dimensional alternatives.

However, the 2-D representation is subject to restricted application. It permits the alignment of two point data sets and is, therefore, applicable to localisation from an *a priori* map, where the map is the reference scan and the robot sensors obtain the observation scan. It might also apply to batch map building, where a set of logged scans are aligned pair-wise to produce a conglomerate off-line map.

But, the 2-D representation is not suitable for applications where the reference scan information is updated by a recursive filter (i.e., the reference scan is non-static). This excludes it from on-line map building and SLAM. Basically, a recursive estimate of the ref-

erence scan will not result in an improved description of the environment but will eventually collapse to a single point location (see Section 4.7 for further discussion).

The on-line mapping and SLAM problems must be represented by a higher-dimensional state-space. For SLAM, the feature-based approach has known practical solutions and the measurement-space concept (without data association) is at least theoretically sufficient. Occupancy grids, on the other hand, are not suitable for SLAM as they do not maintain the necessary correlations between their parameters. For on-line mapping, given location, the feature-based and measurement-space methods are again applicable, and the occupancy grid approach is also reasonable, as the correspondence heuristic between grid cells is sufficient to produce a valid and effective map.

4.3 A Bayesian Likelihood Function for a Point Target Model

This section derives a likelihood function for scan correlation based on the point target model. It shows that, given two scan PDFs, each representing the distribution of a single point target, the likelihood function for their relative pose is computed by their cross-correlation.

As a concrete illustration of scan correlation, this discussion is presented in terms of robot localisation in a plane. Therefore, scan correlation becomes the alignment of an uncertain observation to an uncertain map, where the map contains a single point landmark. (Performing localisation in this manner is subject to a minor caveat. After alignment of the first observation scan, the map and the robot pose become correlated and should no longer be treated as independent when aligning subsequent observations. However, the inconsistency due to this correlation is easily absorbed by adding process stabilising noise, which means that the map can be treated as independent in practice.)

4.3.1 Notation

This section uses some specialised notation as follows. Given a 2-D random vector $\mathbf{x} \in \mathcal{X}$, its PDF is defined by the *distribution function* $f_{\mathbf{x}}(x, y)$. The subscript denotes that this function represents a PDF of \mathbf{x} , and the scalar variables x and y are independent parameters over the space \mathcal{X} . If two random vectors \mathbf{a} and \mathbf{b} are concatenated as $[\mathbf{a}, \mathbf{b}]^T$, their distribution function is given by $f_{\mathbf{ab}}(\cdot)$. For example, if \mathbf{a} and \mathbf{b} were 2-D and 1-D, respectively, it would be $f_{\mathbf{ab}}(x, y, z)$.

The state at time k is the robot pose $\mathbf{x}_k = [x_k, y_k, \phi_k]^T$. The likelihood function for the state is denoted $\Lambda(\mathbf{x}_k) = f_{\mathbf{x}_k}(x, y, \phi)$.

4.3.2 Robot Localisation in a Plane

In a plane, the robot pose is composed of three variables (x_k, y_k, ϕ_k) . However, given a single point-location landmark, as shown in Figure 4.2, it is possible to solve for only two of these states, and therefore necessary to fix one of them to find a solution to the other two. In this presentation, the heading value is assumed known such that $\phi_k = \phi_o$. The exposition below shows that, for any fixed ϕ_k , the likelihood function for the robot pose

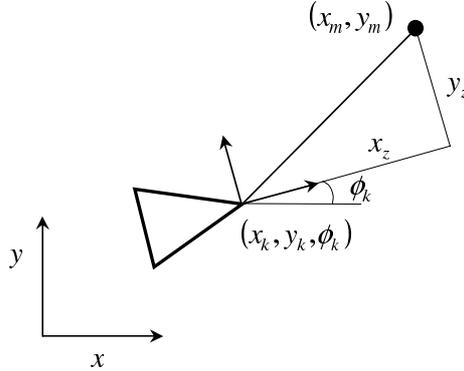


Figure 4.2: Two-dimensional robot localisation. The single point landmark is stored in the map as (x_m, y_m) , and observed from the vehicle by the measurement (x_z, y_z) .

is given by the cross-correlation of the observation and map PDFs, and argues that this is also the case if ϕ_k varies.

In Figure 4.2, the map defines the landmark location by $\mathbf{x}_m = [x_m, y_m]^T$ and the vehicle-centred observation measures this landmark as $\mathbf{x}_z = [x_z, y_z]^T$. Both \mathbf{x}_m and \mathbf{x}_z are random variables with PDFs represented by $f_{\mathbf{m}}(x, y)$ and $f_{\mathbf{z}}(x, y)$, respectively (note that, $f_{\mathbf{m}}(0, 0)$ denotes the map density at the map coordinate origin, and $f_{\mathbf{z}}(0, 0)$ is the measurement density at the vehicle coordinate origin). The vehicle state is $\mathbf{x}_k = [x_k, y_k]^T$ with a known heading ϕ_o .

The following derivation uses the method for transforming probability densities shown in [109, pages 146 and 173].⁴ Given the independent PDFs $f_{\mathbf{m}}(x, y)$ and $f_{\mathbf{z}}(x, y)$, it is desired to find the state PDF $f_{\mathbf{x}_k}(x, y)$ (which is simply the likelihood function $\Lambda(\mathbf{x}_k)$ for fixed ϕ_k). The state is related to the observation by the following equation.

$$\mathbf{x}_k = \mathbf{x}_m - \mathbf{R}\mathbf{x}_z \quad (4.1)$$

where \mathbf{R} is the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos \phi_o & -\sin \phi_o \\ \sin \phi_o & \cos \phi_o \end{bmatrix} \quad (4.2)$$

Using the auxiliary variables a_1 and a_2 to provide four equations in four unknowns, Equation 4.1 becomes

$$\begin{aligned} x_k &= x_m - x_z \cos \phi_o + y_z \sin \phi_o \\ y_k &= y_m - x_z \sin \phi_o - y_z \cos \phi_o \\ a_1 &= x_m \\ a_2 &= y_m \end{aligned} \quad (4.3)$$

which is referred to functionally as $(x_k, y_k, a_1, a_2) = \mathbf{g}(x_z, y_z, x_m, y_m)$. This has the unique

⁴Thanks to Jose Guivant and Hugh Durrant-Whyte for their assistance with this derivation.

solution

$$\begin{aligned} x_z &= (a_1 - x_k) \cos \phi_o + (a_2 - y_k) \sin \phi_o \\ y_z &= -(a_1 - x_k) \sin \phi_o + (a_2 - y_k) \cos \phi_o \\ x_m &= a_1 \\ y_m &= a_2 \end{aligned} \quad (4.4)$$

The joint probability distribution for the transformation in Equation 4.3 is obtained as follows (see [109, page 173]).

$$f_{\mathbf{x}_k \mathbf{a}}(x_k, y_k, a_1, a_2) = \frac{f_{\mathbf{z} \mathbf{m}}(x_z, y_z, x_m, y_m)}{\text{abs}(|\nabla \mathbf{g}_{z \mathbf{m}}|)} \quad (4.5)$$

where the Jacobian $\nabla \mathbf{g}_{z \mathbf{m}}$ is given by

$$\nabla \mathbf{g}_{z \mathbf{m}} = \frac{\partial \mathbf{g}}{\partial (x_z, y_z, x_m, y_m)} = \begin{bmatrix} -\cos \phi_o & \sin \phi_o & 1 & 0 \\ -\sin \phi_o & -\cos \phi_o & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

The determinant of $\nabla \mathbf{g}_{z \mathbf{m}}$ is one and, since \mathbf{x}_z and \mathbf{x}_m are independent, Equation 4.5 becomes

$$f_{\mathbf{x}_k \mathbf{a}}(x_k, y_k, a_1, a_2) = f_{\mathbf{z}}(x_z, y_z) f_{\mathbf{m}}(x_m, y_m) \quad (4.7)$$

where the values of x_z , y_z , x_m , and y_m are the particular solutions found in Equation 4.4. The PDF $f_{\mathbf{x}_k}(x_k, y_k)$ is extracted from this result by integrating over all (a_1, a_2) (see [109, page 125]).

$$f_{\mathbf{x}_k}(x_k, y_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{\mathbf{z}}(b_1, b_2) f_{\mathbf{m}}(a_1, a_2) da_1 da_2 \quad (4.8)$$

where b_1 and b_2 abbreviate the solutions for x_z and y_z , respectively.

$$\begin{aligned} b_1 &= (a_1 - x_k) \cos \phi_o + (a_2 - y_k) \sin \phi_o \\ b_2 &= (x_k - a_1) \sin \phi_o + (a_2 - y_k) \cos \phi_o \end{aligned}$$

Intuitively, Equation 4.8 is interpreted as first rotating the observation PDF $f_{\mathbf{z}}(x, y)$ by ϕ_o and then performing two-dimensional cross-correlation with the map PDF. This can be shown more clearly by defining the rotated observation $\mathbf{x}_r = \mathbf{R} \mathbf{x}_z$. Plainly, \mathbf{x}_z has the unique solution $\mathbf{x}_z = \mathbf{R}^{-1} \mathbf{x}_r$ and, following the same procedure as demonstrated in Equation 4.5, the PDF of \mathbf{x}_r is

$$f_{\mathbf{r}}(x, y) = f_{\mathbf{z}}(x \cos \phi_o + y \sin \phi_o, -x \sin \phi_o + y \cos \phi_o) \quad (4.9)$$

Therefore, Equation 4.8 may be rewritten as

$$f_{\mathbf{x}_k}(x_k, y_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{\mathbf{r}}(a_1 - x_k, a_2 - y_k) f_{\mathbf{m}}(a_1, a_2) da_1 da_2 \quad (4.10)$$

This result demonstrates that, for any fixed ϕ_k , the likelihood function $\Lambda(\mathbf{x}_k)$ is calculated by the cross-correlation of the map and the rotated observation distributions. Furthermore, for any fixed ϕ_k , the total probability mass of $\Lambda(\mathbf{x}_k)$ is one and, therefore, the probability masses for any two values of ϕ_k are equal. Thus, by rule of proportionality, it follows that the set of cross-correlations for all ϕ_k constitutes a valid likelihood function over the domain of \mathbf{x}_k as ϕ_k varies.

4.4 Sum of Gaussians Scan Correlation

This section presents the conversion of a set of point measurements to a Gaussian sum PDF, and derives the equations for the cross-correlation of two such PDFs. This discussion is given in the context of a laser sensor, which observes scans of range-bearing measurements swept in a 2-D plane.

The following topics are examined with regard to implementing Gaussian sum based scan correlation.

- The equations for converting a scan of range-bearing measurements to a Cartesian Gaussian sum are presented.
- Two possible scaling criteria for the individual Gaussians within the Gaussian sum are proposed, with *equal height* scaling being deemed the more suitable representation.
- The cross-correlation equations for two Gaussian sum PDFs are derived, both for the 1-D and planar cases.
- A second correlation method is presented that treats the measurements in the observation scan individually rather than as a single entity.
- The likelihood functions produced by the two scan correlation methods from this section are compared with the likelihood function resulting from explicit point-wise data association.

4.4.1 Conversion to Gaussian Sum Representation

An n -dimensional Gaussian distribution, with mean value $\bar{\mathbf{p}}$ and covariance \mathbf{P} , is defined by the following equation.

$$g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) \triangleq \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{P}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{p}})\right) \quad (4.11)$$

The integral (or, informally, volume) of this function over the space \mathbb{R}^n is one and, therefore, a *scaled* Gaussian $\alpha g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})$ has a volume equal to α .

An n -dimensional sum of Gaussians PDF is defined as the sum of k scaled Gaussians,

$$G(\mathbf{x}) \triangleq \sum_{i=1}^k \alpha_i g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i) \quad (4.12)$$

where the sum of the scaling factors α_i equals one (i.e., the total probability mass is one). For the purpose of cross-correlation, the correct total probability is not relevant—only relative scale is important. Therefore, in this thesis, the Gaussian sums do not require normalisation (i.e., the α_i 's need not sum to one).

Example 4.1

Conversion of a laser scan to a sum of Gaussians. For a scan of k range-bearing measurements, each measurement $\mathbf{z}_i = (r_i, \theta_i)$ is first converted to sensor-centred Cartesian space.

$$\hat{\mathbf{x}}_i = \mathbf{f}(\mathbf{z}_i) = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{bmatrix}$$

$$\mathbf{P}_i = \nabla \mathbf{f}_{\mathbf{z}_i} \mathbf{R}_i \nabla \mathbf{f}_{\mathbf{z}_i}^T$$

where the Jacobian $\nabla \mathbf{f}_{\mathbf{z}_i}$ is given by

$$\nabla \mathbf{f}_{\mathbf{z}_i} = \frac{\partial \mathbf{f}}{\partial \mathbf{z}_i} = \begin{bmatrix} \cos \theta_i & -r_i \sin \theta_i \\ \sin \theta_i & r_i \cos \theta_i \end{bmatrix}$$

These Gaussian estimates are then compiled as a Gaussian sum as follows.

$$G(\mathbf{x}) = \sum_{i=1}^k \frac{\alpha_i}{2\pi \sqrt{|\mathbf{P}_i|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_i)^T \mathbf{P}_i^{-1}(\mathbf{x} - \hat{\mathbf{x}}_i)\right)$$

A complete theoretical basis for the values of the α_i 's is not presently known. However, two basic scaling criteria have been trialed experimentally, and the effects of these values on the scan correlation results are discussed below.

4.4.2 Scaling Models for Gaussian Sum

An intuitive summation of the individual measurement Gaussians is to simply add them without scaling.

$$G(\mathbf{x}) = \sum_{i=1}^k g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i) \quad (4.13)$$

However, it transpires that this formulation gives a skewed representation of the environment, and it would seem that a more suitable scaling is possible. To illustrate, Figure 4.3 shows a scan of laser measurements transformed to Cartesian space, and Figure 4.4(a) depicts the Gaussian sum PDF for this scan, based on Equation 4.13. Since, each unscaled Gaussian has equal (unit) volume, those with small uncertainty⁵ possess very high peaks, indicating high likelihood. The effect of this scaling model on scan correlation is that the short (highly certain) measurements swamp the cross-correlation result, and more distance points have little influence.

The difficulty with scaling arises from the discrepancy between the physical reality of the sensed measurements (i.e., discrete points on a surface) and the interpretation of the Cartesian parameter-space model (i.e., distribution of a single point target). The intuitive “equal volume” scaling model makes sense if each measurement actually does observe the point target, as more accurate measurement would give a better estimate of its location. However, the individual Gaussians in the Gaussian sum must be understood as defining

⁵For the laser sensor, the measurement uncertainties in range and bearing are assumed constant. Therefore, the uncertainty in Cartesian space increases with distance from the observer.

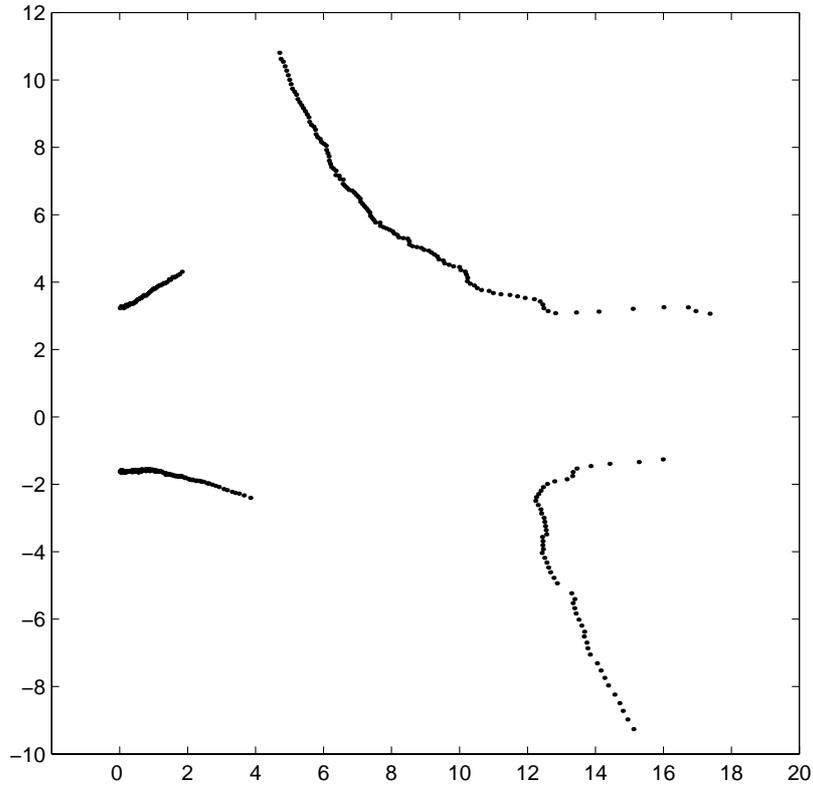


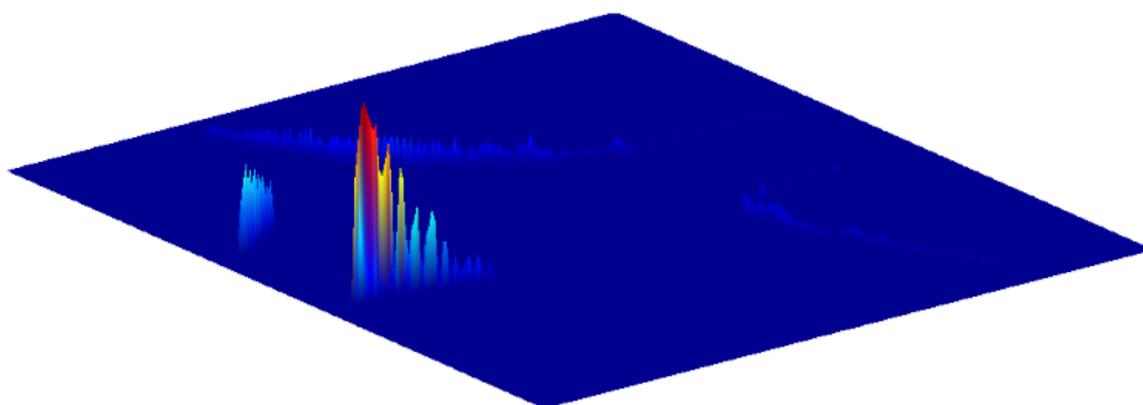
Figure 4.3: Unprocessed laser scan measurements of a mine tunnel wall.

possible locations for the target, and a more accurate measurement is no more likely than a coarse measurement. In fact, the inaccurate measurement, covering a wider region of the state-space, actually conveys *more* information about the target’s possible location. (This is clear from the beam-width of the laser sensor; an accurate near measurement detects the likelihood of a target within a very small region, and so provides little information about the environment, while a more distant measurement covers a broader spatial region.)

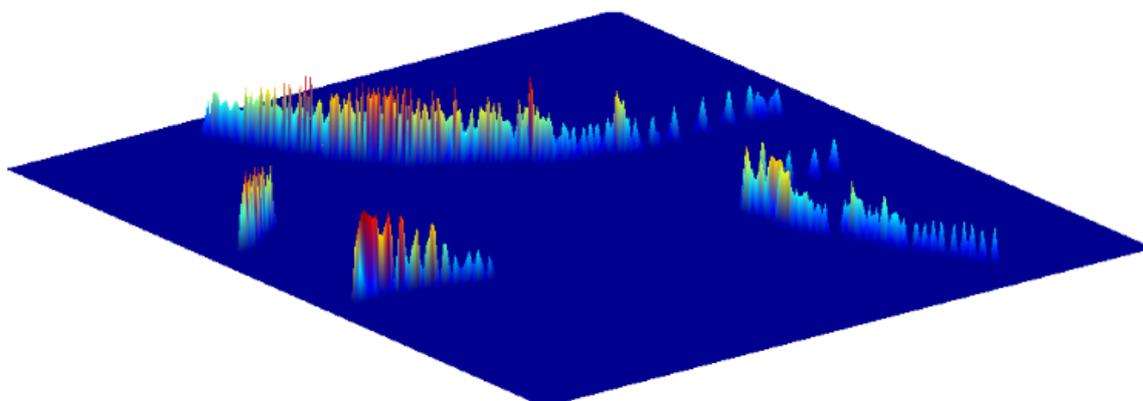
An alternative scaling model is to make each Gaussian equal height.

$$G(\mathbf{x}) = \sum_{i=1}^k \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{p}}_i)^T \mathbf{P}_i^{-1}(\mathbf{x} - \bar{\mathbf{p}}_i)\right) \quad (4.14)$$

In other words, each Gaussian has a scaling factor $\alpha_i = (2\pi)^{n/2} \sqrt{|\mathbf{P}_i|}$. The Gaussian sum PDF for the example scan is shown in Figure 4.4(b), and depicts a much more even spatial representation. This “equal height” scaling model asserts that the point target is equally likely to exist within the distribution of an accurate or inaccurate measurement, but that the coarser measurement covers a wider spatial region and so conveys more information; thus, the wider Gaussians have greater volume.



(a) Equal volume Gaussians



(b) Equal height Gaussians

Figure 4.4: Two alternative Gaussian sum representations for the laser scan in Figure 4.3. The first assigns each point measurement a Gaussian of equal volume. This results in very high peaks for measurements with low uncertainty. The second assigns Gaussians of equal height, which stipulates that very precise measurements contain less information about the shape of the environment (i.e., smaller volume Gaussians), and produces a better spatial description.

4.4.3 Cross-Correlation in One Dimension

The likelihood function for correlating two Gaussian sum PDFs is derived here for the n -dimensional case, but is strictly correct only for the 1-D case. That is, this form is valid for higher dimensions only if the angular alignment (e.g., roll, pitch, yaw) between the two data sets is fixed.

As preliminary information, the following two points are noted. First, the volume under a scaled Gaussian is equal to the scale factor.

$$\begin{aligned} \int_{-\infty}^{\infty} \alpha g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) d\mathbf{x} &= \alpha \int_{-\infty}^{\infty} g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) d\mathbf{x} \\ &= \alpha \end{aligned} \quad (4.15)$$

Second, the volume obtained from the multiplication of two Gaussians

$$\int_{-\infty}^{\infty} g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q}) d\mathbf{x} = \int_{-\infty}^{\infty} \alpha g(\mathbf{x}; \bar{\mathbf{r}}, \mathbf{R}) d\mathbf{x} \quad (4.16)$$

is equal to the following (see Appendix E for derivation).

$$\alpha = \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{P} + \mathbf{Q}|}} \exp\left(-\frac{1}{2}(\bar{\mathbf{p}} - \bar{\mathbf{q}})^T (\mathbf{P} + \mathbf{Q})^{-1} (\bar{\mathbf{p}} - \bar{\mathbf{q}})\right) \quad (4.17)$$

Therefore, given two Gaussian sums

$$\begin{aligned} G_o(\mathbf{x}) &= \sum_{i=1}^{k_1} \alpha_i g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i) \\ G_r(\mathbf{x}) &= \sum_{i=1}^{k_2} \beta_i g(\mathbf{x}; \bar{\mathbf{q}}_i, \mathbf{Q}_i) \end{aligned} \quad (4.18)$$

the likelihood function (i.e., cross-correlation) is calculated as follows.

$$\begin{aligned} \Lambda(\mathbf{x}_k) &= G_o(\mathbf{x}_k) \star G_r(\mathbf{x}_k) \\ &= \int_{-\infty}^{\infty} \sum_{i=1}^{k_1} \alpha_i g(\mathbf{u} - \mathbf{x}_k; \bar{\mathbf{p}}_i, \mathbf{P}_i) \sum_{j=1}^{k_2} \beta_j g(\mathbf{u}; \bar{\mathbf{q}}_j, \mathbf{Q}_j) d\mathbf{u} \\ &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_i \beta_j \int_{-\infty}^{\infty} g(\mathbf{u} - \mathbf{x}_k; \bar{\mathbf{p}}_i, \mathbf{P}_i) g(\mathbf{u}; \bar{\mathbf{q}}_j, \mathbf{Q}_j) d\mathbf{u} \\ &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \alpha_i \beta_j \gamma_{ij}(\mathbf{x}_k) \end{aligned} \quad (4.19)$$

where the function $\gamma_{ij}(\mathbf{x}_k)$ is the cross-correlation of the two Gaussians $g(\mathbf{x}; \bar{\mathbf{p}}_i, \mathbf{P}_i)$ and $g(\mathbf{x}; \bar{\mathbf{q}}_j, \mathbf{Q}_j)$. From Equation 4.17, this function is given by

$$\gamma_{ij}(\mathbf{x}_k) = \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{P}_i + \mathbf{Q}_j|}} \exp\left(-\frac{1}{2}(\mathbf{x}_k + \bar{\mathbf{p}}_i - \bar{\mathbf{q}}_j)^T (\mathbf{P}_i + \mathbf{Q}_j)^{-1} (\mathbf{x}_k + \bar{\mathbf{p}}_i - \bar{\mathbf{q}}_j)\right) \quad (4.20)$$

Algorithm 4.2: LIKELIHOOD(G_o, G_r, \mathbf{x}_k)

```

 $\Lambda \leftarrow 0$ 
 $\mathbf{t} \leftarrow \begin{bmatrix} x_k \\ y_k \end{bmatrix}$ 
 $\mathbf{R} \leftarrow \begin{bmatrix} \cos \phi_k & -\sin \phi_k \\ \sin \phi_k & \cos \phi_k \end{bmatrix}$ 
for  $i \leftarrow 1$  to  $|G_o|$ 
   $\mathbf{p}' \leftarrow \mathbf{t} + \mathbf{R}\bar{\mathbf{p}}_i$ 
   $\mathbf{P}' \leftarrow \mathbf{R}\mathbf{P}_i\mathbf{R}^T$ 
  for  $j \leftarrow 1$  to  $|G_r|$ 
    do  $\left\{ \begin{array}{l} \nu \leftarrow \mathbf{p}' - \bar{\mathbf{q}}_j \\ \mathbf{S} \leftarrow \mathbf{P}' + \mathbf{Q}_j \\ \gamma \leftarrow \frac{1}{2\pi\sqrt{|\mathbf{S}|}} \exp\left(-\frac{1}{2}\nu^T\mathbf{S}^{-1}\nu\right) \\ \Lambda \leftarrow \Lambda + \alpha_i\beta_j\gamma \end{array} \right.$ 
return ( $\Lambda$ )

```

4.4.4 Planar Cross-Correlation

For the planar case, such as the 2-D robot localisation problem, the state $\mathbf{x}_k = (x_k, y_k, \phi_k)$ varies in ϕ_k and so $\Lambda(\mathbf{x}_k)$ cannot be obtained directly by cross-correlation. However, given the result in Section 4.3.2, it is sufficient to alter Equation 4.20 as follows,

$$\gamma_{ij}(\mathbf{x}_k) = \frac{1}{2\pi\sqrt{|\mathbf{P}' + \mathbf{Q}_j|}} \exp\left(-\frac{1}{2}(\mathbf{p}' - \bar{\mathbf{q}}_j)^T(\mathbf{P}' + \mathbf{Q}_j)^{-1}(\mathbf{p}' - \bar{\mathbf{q}}_j)\right) \quad (4.21)$$

where \mathbf{p}' and \mathbf{P}' are the mean and covariance of the i th Gaussian in the observation PDF transformed to the reference coordinate frame,

$$\begin{aligned} \mathbf{p}' &= \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \mathbf{R}\bar{\mathbf{p}}_i \\ \mathbf{P}' &= \mathbf{R}\mathbf{P}_i\mathbf{R}^T \\ \mathbf{R} &= \begin{bmatrix} \cos \phi_k & -\sin \phi_k \\ \sin \phi_k & \cos \phi_k \end{bmatrix} \end{aligned}$$

Algorithm 4.2 provides a simple implementation of the planar likelihood function (where $|G|$ denotes the number of Gaussians in a Gaussian sum G). For two Gaussian sums of size k_1 and k_2 , this algorithm has complexity $O(k_1k_2)$ and, for larger PDF definitions, is prohibitively slow for estimation methods like particle filtering. However, assuming the Gaussians in each sum possess reasonably similar covariance and scale, the added likelihood due to two particular Gaussians becomes negligible if the two-norm $\|\mathbf{p}' - \bar{\mathbf{q}}_j\|$ is large. This leads to the possibility of various optimisations, two of which are discussed below.

First, the search space (in the reference Gaussian sum) can be partitioned so that, for a given mean \mathbf{p}' , only those Gaussians satisfying $\|\mathbf{p}' - \bar{\mathbf{q}}_j\| < d$ are incorporated into the likelihood calculation. This might be achieved by binning the mean values $\bar{\mathbf{q}}_j$ in a coarse grid of granularity $2d$. Alternatively, a sufficiently accurate result might be obtained by using only the k nearest neighbours to \mathbf{p}' . This method, implemented with k-d trees [12, 101], is used in the experimental applications of this chapter.

A second optimisation is to cache a set of near neighbours for each Gaussian in G_o , and retain these while ever the state \mathbf{x}_k is reasonably close to its initial value. This is particularly useful for maximum likelihood estimation, which will converge even if the near neighbour sets are only updated every n iterations.

4.4.5 An Alternative Correlation Scheme

In the previous discussion, the reference and observation scans are both represented as a Gaussian sum PDFs, but an alternative proposition is to consider each measurement in the observation scan as an independent observation of the reference PDF. Thus, each measurement determines a separate likelihood function

$$\Lambda_i(\mathbf{x}_k) = \sum_{j=1}^{k_2} \beta_j \gamma_{ij}(\mathbf{x}_k) \quad (4.22)$$

where $\gamma_{ij}(\mathbf{x}_k)$ is given in Equation 4.21. Notice that the scaling factor α_i for each observation Gaussian is not required, since each measurement forms a unit volume PDF. The combined likelihood for the observation scan, therefore, is the intersection of the individual likelihoods.

$$\Lambda(\mathbf{x}_k) = \prod_{i=1}^{k_1} \Lambda_i(\mathbf{x}_k) \quad (4.23)$$

This correlation method, with due assumptions, is shown in Section 4.4.6 to produce more optimal likelihood results than is obtained with a single observation PDF, but there are several complicating issues.

First, the reference PDF is reused for each iteration of Equation 4.22. This correlation between the likelihoods might be compensated by inflating the observation uncertainty but, since each observation usually observes only a small portion of the reference PDF (and recalling that the reference PDF is itself composed of independent measurements), it is difficult to determine the quantity of this inflation. A better solution, perhaps, is to use the correlated reference information without inflation, given the understanding that Equation 4.23 will converge to a best fit of the reference PDF and not the true state of the environment.

A second problem is that the observation scan not only observes parts of the reference scan, but may also include additional measurements due to viewpoint variation, occlusion and dynamic objects; corporately these are termed *outlier* measurements. The presence of outliers means that a state value \mathbf{x}_k close to the true state may be made unlikely by the multiplication of $\Lambda_i(\mathbf{x}_k)$, where i represents an outlier measurement. This is particularly so if the measurements possess bounded uncertainty, in which case the probability of the true state may become zero. A work-around for this problem is to specify a probability of

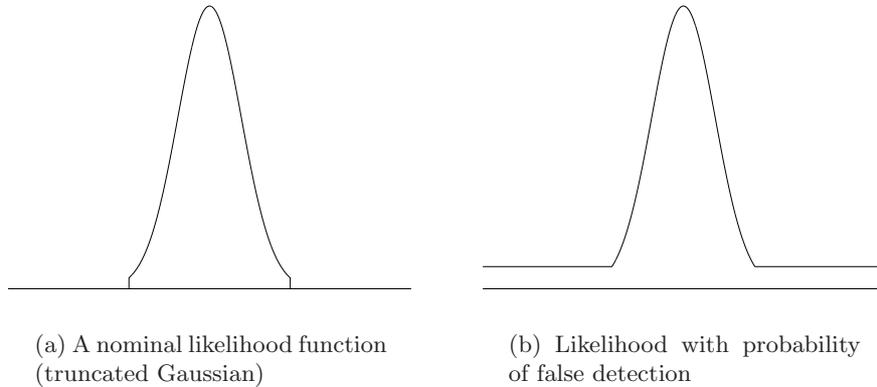


Figure 4.5: Likelihood function with probability of false detection. Any value $\Lambda(\mathbf{x}_k)$ less than \mathcal{P}_f becomes equal to \mathcal{P}_f .

false detection \mathcal{P}_f to set a minimum likelihood for any state value (see Figure 4.5). In other words, given a particular state $\mathbf{x}_k = \mathbf{x}_0$, if $\Lambda_i(\mathbf{x}_0)$ is less than \mathcal{P}_f , then the measurement i is assumed an outlier for \mathbf{x}_0 . Determining an appropriate value for \mathcal{P}_f is a matter of tuning.⁶

This method has not yet been implemented and is not used in the experimental applications below; all results using real data are obtained with the more conservative representation of a single observation scan PDF.

4.4.6 A Comparison of Correlation and Explicit Data Association

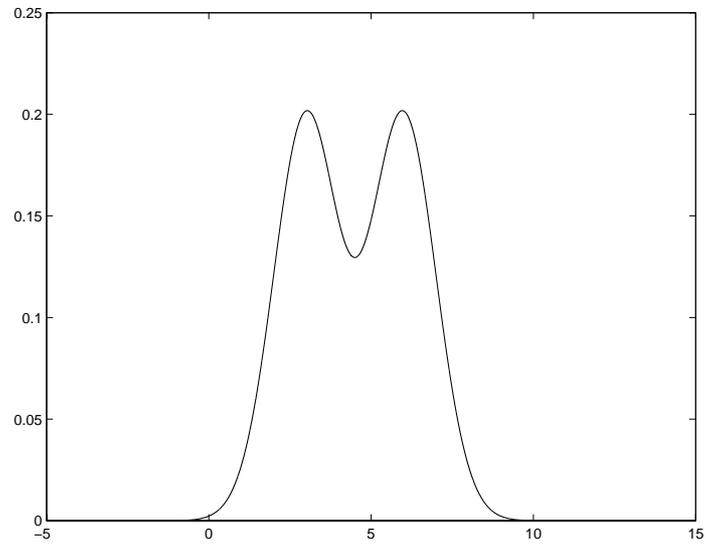
The two correlation methods described above are demonstrated via a simple 1-D example, and their results are compared to the likelihood function obtained from explicit data association (i.e., where each observation measurement is assigned to a particular reference landmark).

The example depicts a simple case where the environment consists of two point landmarks (e.g., two trees), and the reference scan represents them by two Gaussians of equal uncertainty, and the observation scan is exactly the same as the reference. Two variations of this example are presented: the first where the two Gaussians are merged (see Figure 4.6(a)) and the second where they are distinct (see Figure 4.7(a)). These instances serve to compare the characteristics of the three association methods.

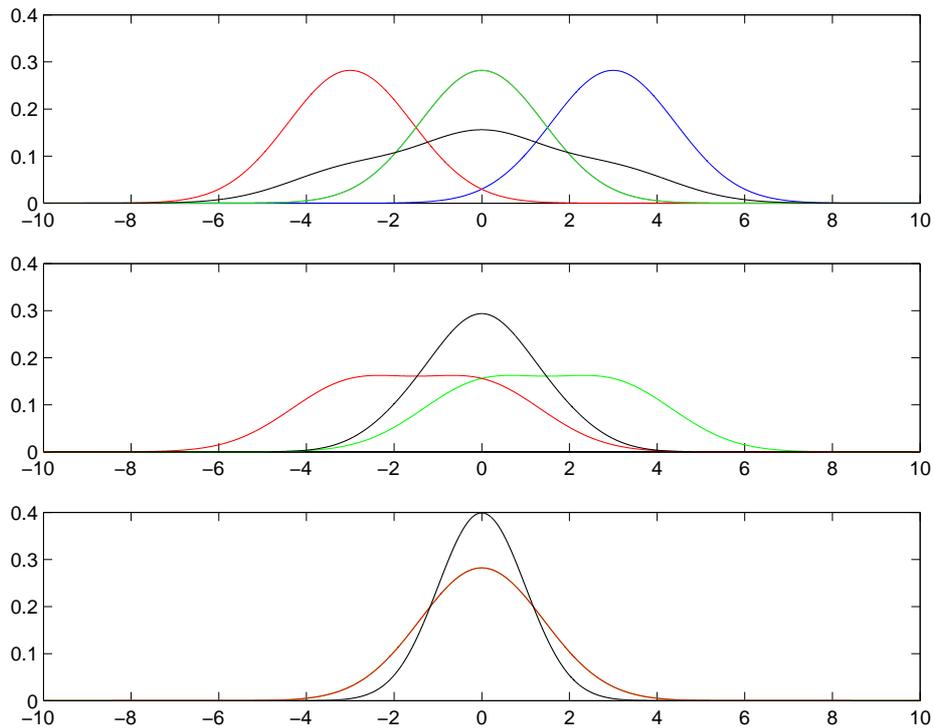
To begin with, the equations for the three versions of likelihood function are shown. Each share the basic calculation of the correlation of two Gaussians where, from Equation 4.20, the cross-correlation of two 1-D Gaussians is given by

$$\alpha_{ij}(x_k) = \frac{1}{\sqrt{2\pi(\sigma_i^2 + \sigma_j^2)}} \exp\left(-\frac{(x_k + \bar{x}_i - \bar{x}_j)^2}{2(\sigma_i^2 + \sigma_j^2)}\right) \quad (4.24)$$

⁶One important factor in the tuning of \mathcal{P}_f is the local density of the reference PDF, which is dependent on the total size of the reference scan. For a reference scan built from laser data, this difficulty may be circumvented by not normalising the reference Gaussian sum. Other influences on \mathcal{P}_f may perhaps be subsumed by empirical tuning.

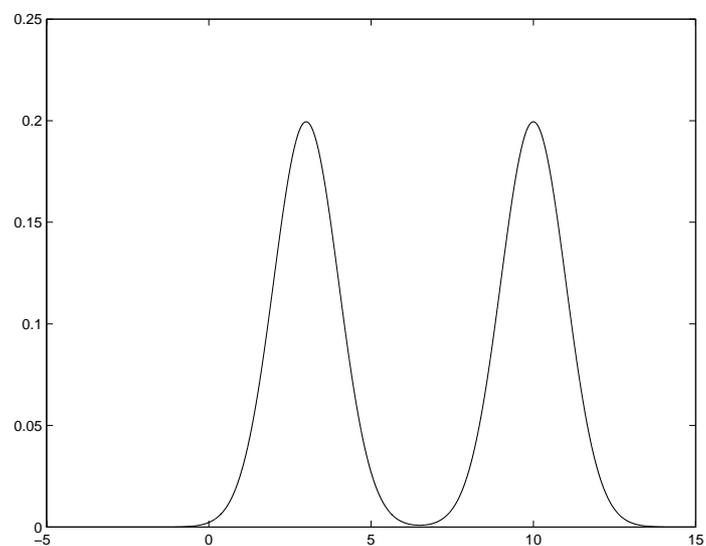


(a) Scan Gaussian sum

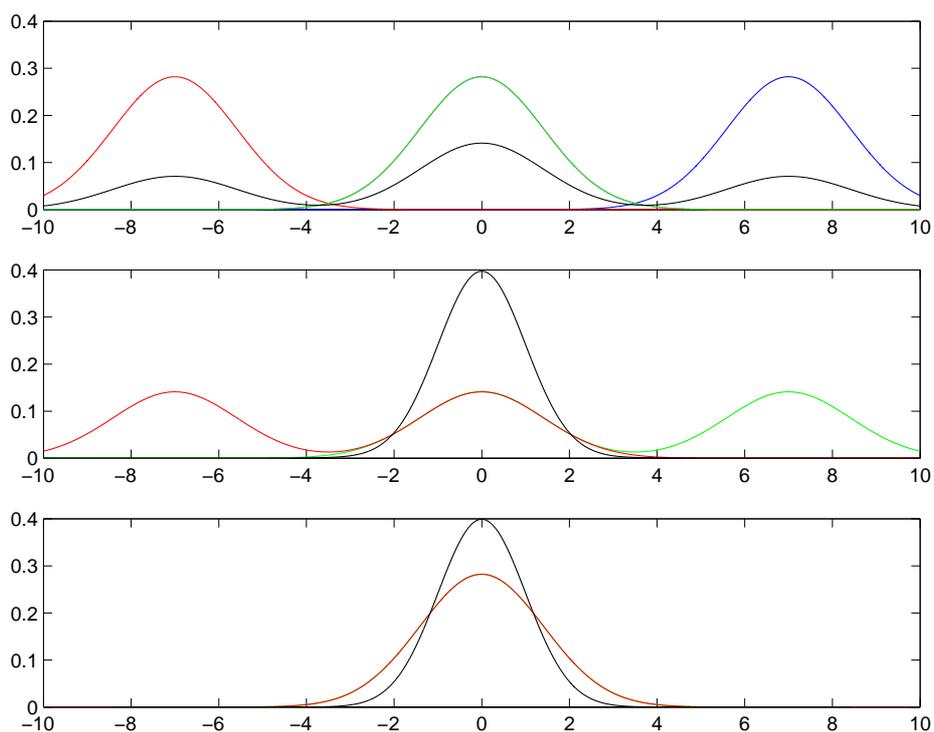


(b) Likelihood functions

Figure 4.6: Correlation results for merged measurements. The measurement Gaussians (a) are merged, resulting in the likelihood functions shown in (b). Note, the constituent α_{ij} 's are shown in colour, while the resultant likelihood functions are black.



(a) Scan Gaussian sum



(b) Likelihood functions

Figure 4.7: Correlation results for distinct measurements. The measurement Gaussians (a) are well separated, resulting in the likelihood functions shown in (b).

where i represents an observation Gaussian and j a reference Gaussian. In the equations below, this area is denoted α_{ij} , without the implicit state parameter.

The first correlation-based likelihood function is for the case where the observation scan forms a single Gaussian sum PDF. Thus, the reference and observation PDFs each consist of two scaled Gaussians (with scale factors equal to $\frac{1}{2}$).

$$\Lambda(x_k) = \frac{1}{4}(\alpha_{11} + \alpha_{12} + \alpha_{21} + \alpha_{22}) \quad (4.25)$$

Ignoring the (irrelevant) constant term, this is expressed as

$$\Lambda(x_k) = \alpha_{11} + \alpha_{12} + \alpha_{21} + \alpha_{22} \quad (4.26)$$

Notice that this likelihood function is the union of the individual likelihoods.

The second correlation-based likelihood function is obtained when the observation measurements are processed independently according to Section 4.4.5 (n.b., without inflating the observation uncertainties).

$$\Lambda(x_k) = (\alpha_{11} + \alpha_{12})(\alpha_{21} + \alpha_{22}) \quad (4.27)$$

This likelihood function is always less conservative⁷ than Equation 4.26 since it is the intersection of the sums $\alpha_{11} + \alpha_{12}$ and $\alpha_{21} + \alpha_{22}$, which is more constrained than the union of all α 's.

Finally, the third likelihood function (data association) is obtained by defining explicit correspondences, $1 \rightarrow 1$ and $2 \rightarrow 2$, between the observation and reference measurements.

$$\Lambda(x_k) = \alpha_{11}\alpha_{22} \quad (4.28)$$

In other words, this likelihood function is the intersection of the likelihoods for each correspondence, and is clearly the least conservative of the three likelihood functions.

Examining the results of the two examples in Figures 4.6 and 4.7, the reference and (identical) observation Gaussian sums for each example are shown in (a) and their resulting likelihood functions are shown in (b). These results depict (top to bottom) Equations 4.26 to 4.28, respectively. The top result shows the four constituent α_{ij} 's (note, all are Gaussian with equal variance, and α_{11} and α_{22} have equal mean), and their sum; the middle result shows $\alpha_{11} + \alpha_{12}$ and $\alpha_{21} + \alpha_{22}$, and their product; and the bottom result shows α_{11} and α_{22} , and their product. Note, each of these distributions is normalised to possess area equal to one.

In both Figures 4.6 and 4.7, the first correlation method produces a likelihood function that reflects the degree of overlap between the observation and reference PDFs. This can be seen as a conservative likelihood function in the presence of outlier measurements (or missing data etc)—this is particularly apparent in Figure 4.7. The second correlation technique, when compared in the two figures, produces a likelihood function that is more conservative for merged measurements than for discrete measurements. That is, as the measurements

⁷The term *conservative* refers to the shape of a distribution. A flat distribution is totally conservative with all states equally likely; a less conservative PDF is more peaked, with areas of concentrated probability mass.

become more discrete, the product of the α_{12} and α_{21} components approach zero, and Equation 4.27 approaches Equation 4.28. Thus, Equation 4.27 correctly deals with data association uncertainty from merged measurements provided each measurement in the observation scan does, in fact, correspond to something in the reference scan. However, it fails to consider outlier measurements (and so requires a probability of false detection \mathcal{P}_f). Finally, the likelihood function for explicit data association is the same in both figures, as expected. This likelihood function provides greater accuracy than the correlation-based methods if the associations can be guaranteed (i.e., data association constraints, if correct, are a substantial source of information), but is inconsistent if the associations are uncertain (as for merged measurements).

To summarise these results,

- Equation 4.26 is conservative provided *some* of the observation measurements correspond to part of the reference scan.
- Equation 4.27 is conservative provided *all* of the observation measurements correspond to part of the reference scan. To some extent this limitation may be reduced by incorporating a probability for outlier measurements.
- Equation 4.28 is optimal provided the associations are known and correct, otherwise it is optimistic.

4.5 Application: Maximum Likelihood Dead Reckoning

This application uses maximum likelihood scan correlation to perform laser-based dead reckoning. For each scan-pair, the predicted change-in-pose is equal to the maximum likelihood estimate obtained for the previous scan-pair. The estimated change-in-pose is then obtained by searching for the maximum scan cross-correlation using a greedy search method (i.e., the Nelder-Mead downhill simplex method [107, 112]). Note, these results do not supply any measure of uncertainty.

4.5.1 Results

The results for the internal road environment are shown in Figure 4.8. The laser data used here is the same as used for feature-based dead reckoning in the previous chapter.

These results are comparable to the results of the previous chapter but are less accurate for a number of reasons. First, for many scan-pairs the likelihood function was multi-modal and, on several occasions, the greedy optimisation converged to the wrong mode. Second, even when converging to the correct mode, the optimisation step was run for a fixed number of iterations, and may not have converged completely; (the feature-based estimate, on the other hand, is closed form). Third, in some regions there was insufficient common information between scans to allow proper correlation and, while this was detected in the feature-based implementation, it was not detected here and spurious estimates were obtained. (Note, lack of correlation information might be detected by examining the non-normalised value of the maximum cross-correlation.) Finally, the explicit batch data associ-

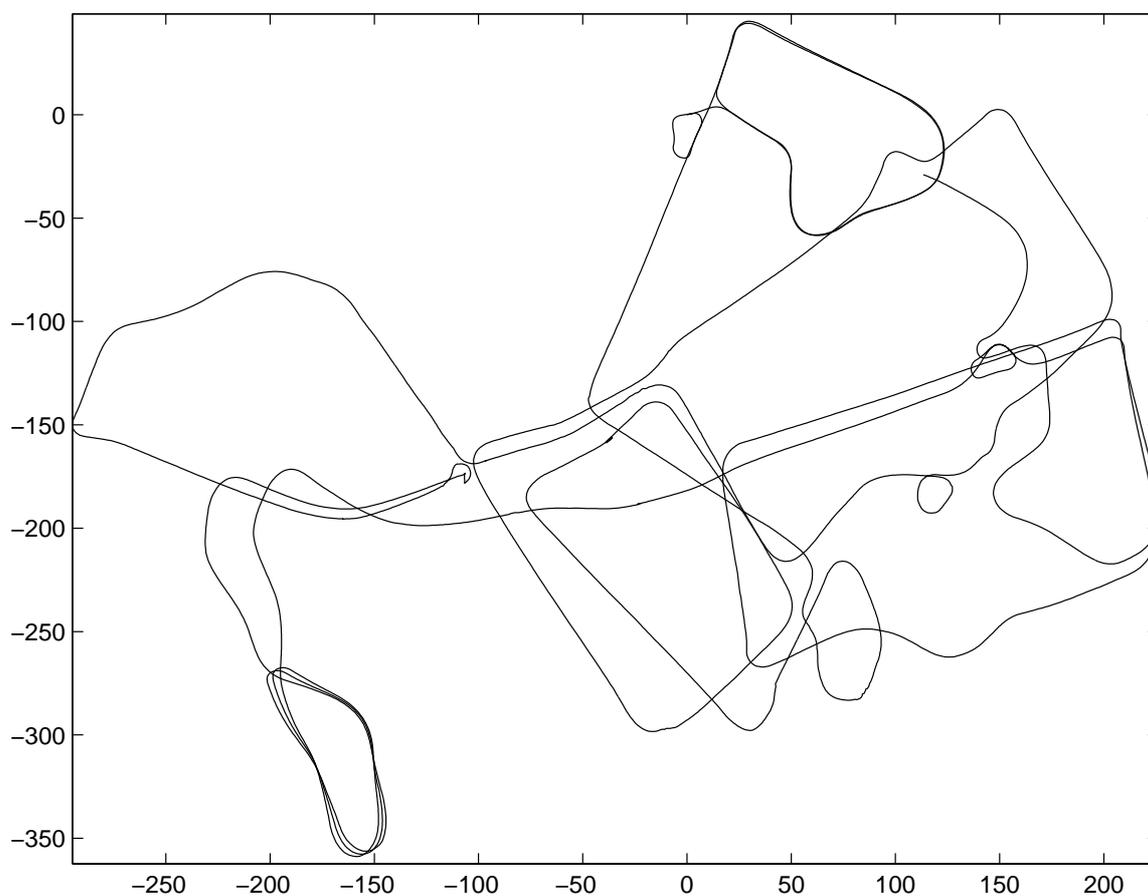


Figure 4.8: Maximum likelihood dead reckoning in the internal road environment. The result here is rather less accurate than was obtained using the feature-based approach.

ation used by the feature-based method imparts substantial information that is unavailable to correlation methods.

The results for the mine tunnel environment are shown in Figure 4.9 (see Appendix A.3 for a description of this environment). In this environment reliable geometric features are difficult to obtain, although some efforts have been made to do so using points of maximum curvature [94]. Nevertheless, the mine walls are rich in information (highly textured), and permit extremely accurate and unambiguous correlation-based estimation. A smooth and precise dead reckoning trajectory is shown; note that the apparent discontinuity at coordinates (72, 2) is, in fact, where the vehicle made a 3-point turn.

In addition to the vehicle trajectory, the unprocessed laser measurements are pictured in Figure 4.9, projected onto their global locations according to the estimated pose of the laser for each scan. This gives a clear indication of the precision of the dead reckoning result as the overlaid points produce walls about 10cm thick—which is plus-minus the laser range resolution (see Figure 4.10).

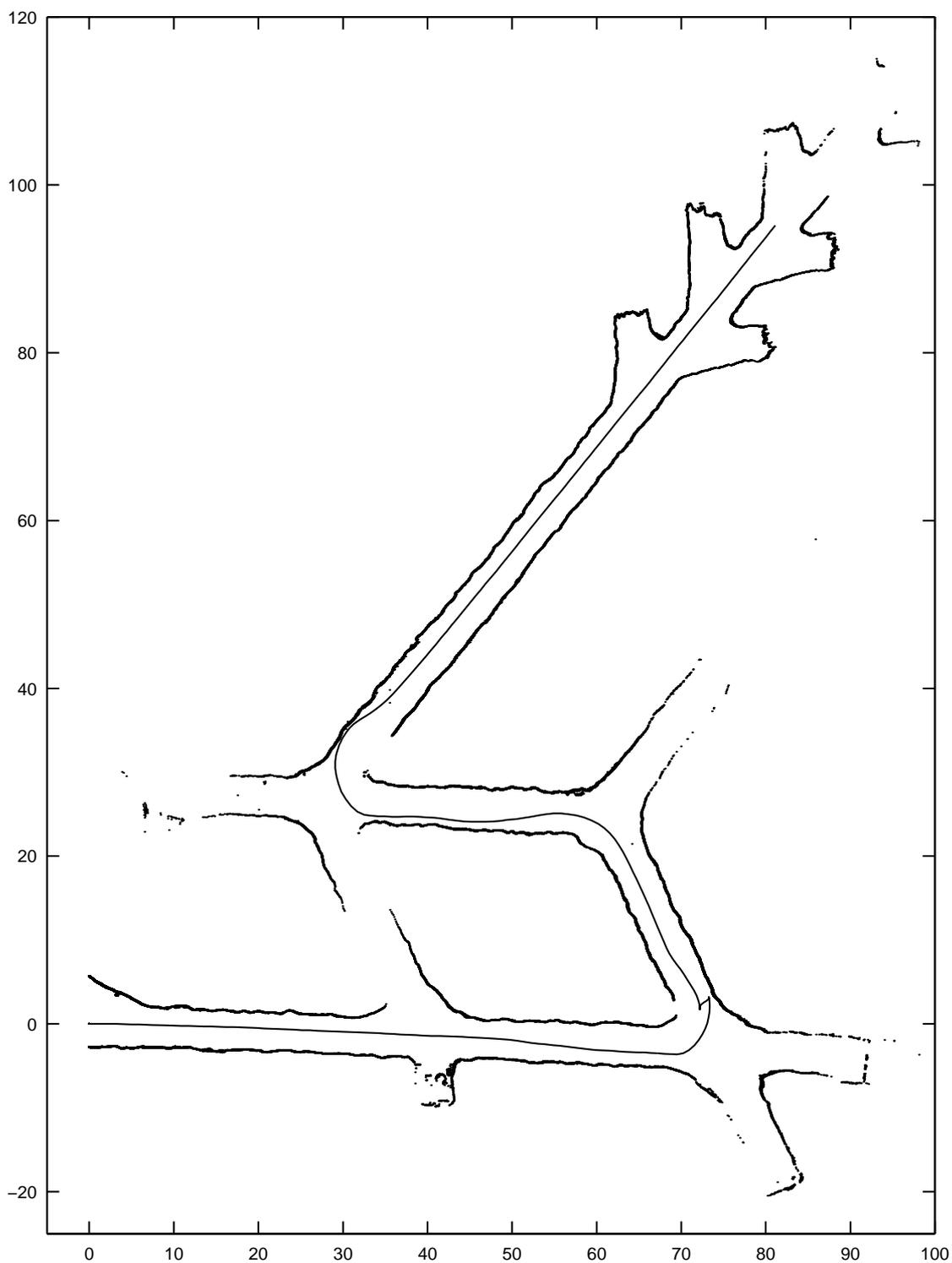


Figure 4.9: Maximum likelihood dead reckoning in the mine environment. Here geometric features are not readily available but excellent results can still be obtained (as demonstrated by the shape of the walls formed by plotting laser points at their predicted global locations).

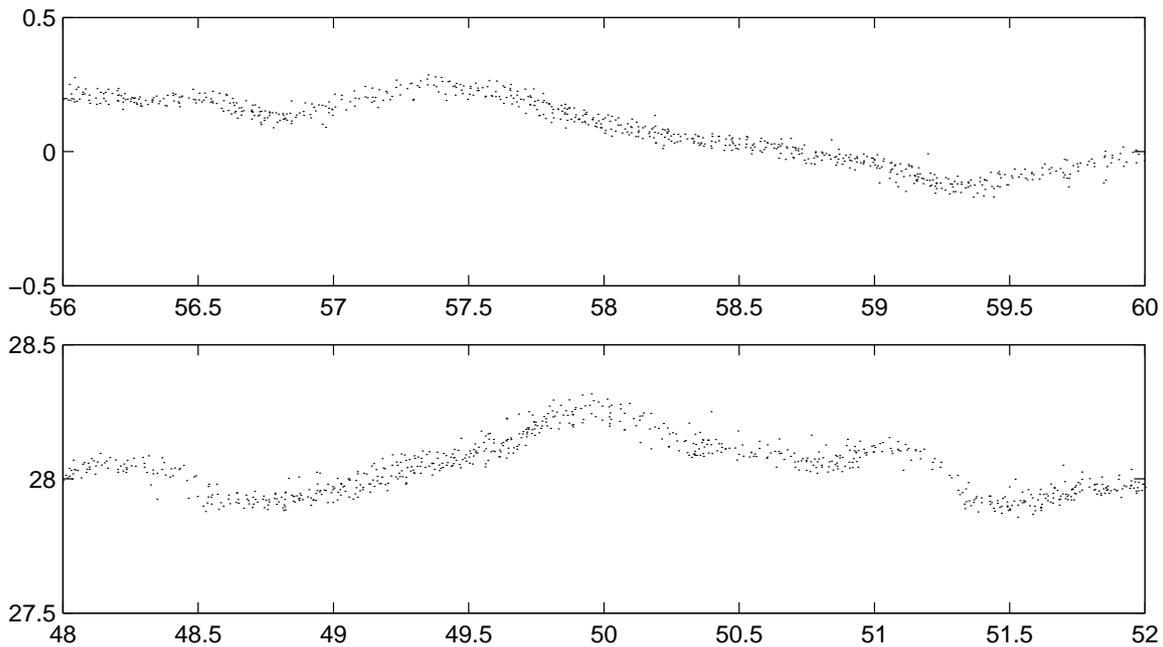


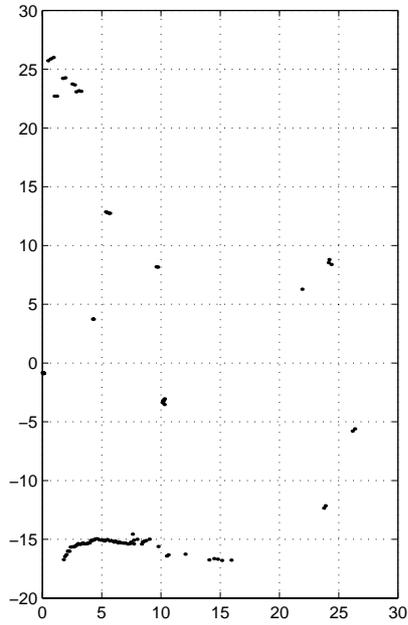
Figure 4.10: Mine walls depicted by unprocessed laser measurements. These two figures are characteristic of the scan correlation dead reckoning result, where the projected laser points form walls with about $\pm 5\text{cm}$ accuracy.

4.5.2 Multimodal Likelihood Functions

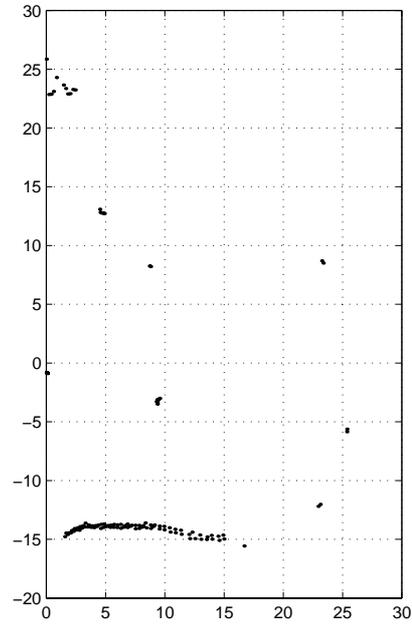
The fragility of maximum likelihood estimation is primarily due to the ambiguity of multimodal likelihood functions. This was true in the internal road environment, where tilting of the laser sensor observed varying cross-sections of the world, including different layers of undergrowth and other clutter.

Two examples of multimodal likelihood functions are shown here from data obtained in the park environment. The first scan-pair, in Figure 4.11, shows a scene containing tree trunk features and also a large section of spurious returns, in the bottom-left corner, due to ground sweeps. The resulting likelihood function shows a narrow peak with mode $(0.83, -0.03, -0.00)$ representing the correct change-in-pose, and a broad peak with mode $(1.48, -1.09, -0.04)$ due to the apparent motion of the ground sweep. Note that the incorrect mode is more likely than the correct one.

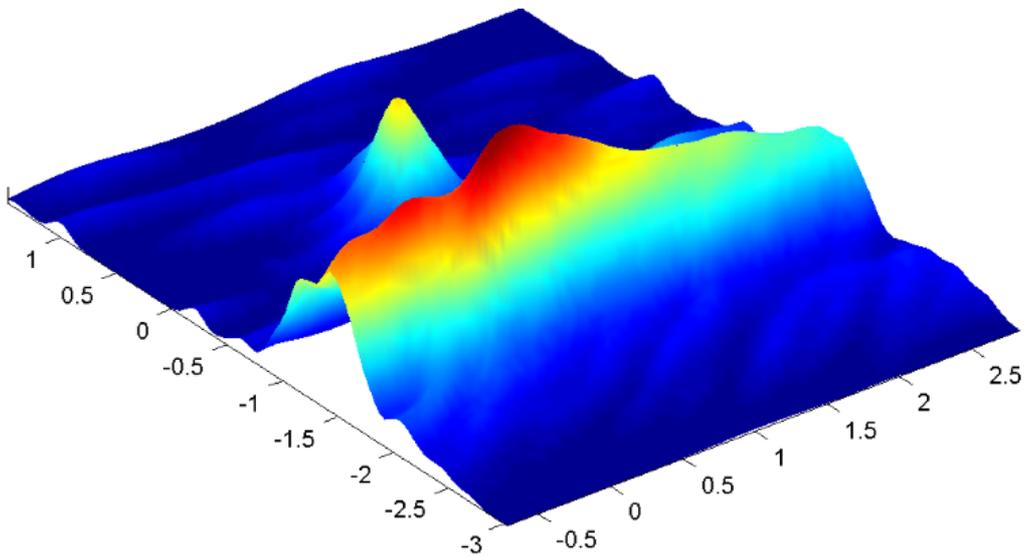
The second scan-pair, in Figure 4.12, again depicts a scene consisting mostly of trees, but this time it also contains a set of returns, in the bottom-left corner, from a moving vehicle. The likelihood function shows a narrow peak at $(0.66, -0.02, 0.00)$ for the true change-in-pose, and a long thin distribution with mode $(-1.70, -1.20, -0.01)$ showing the relative motion of the observer with respect to the (moving) observed vehicle. Once again, the false mode presents the greater likelihood.



(a) Scan 1



(b) Scan 2



(c) Likelihood function

Figure 4.11: Multimodal likelihood function due to ground sweep. Notice that the maximum mode is not the correct mode.

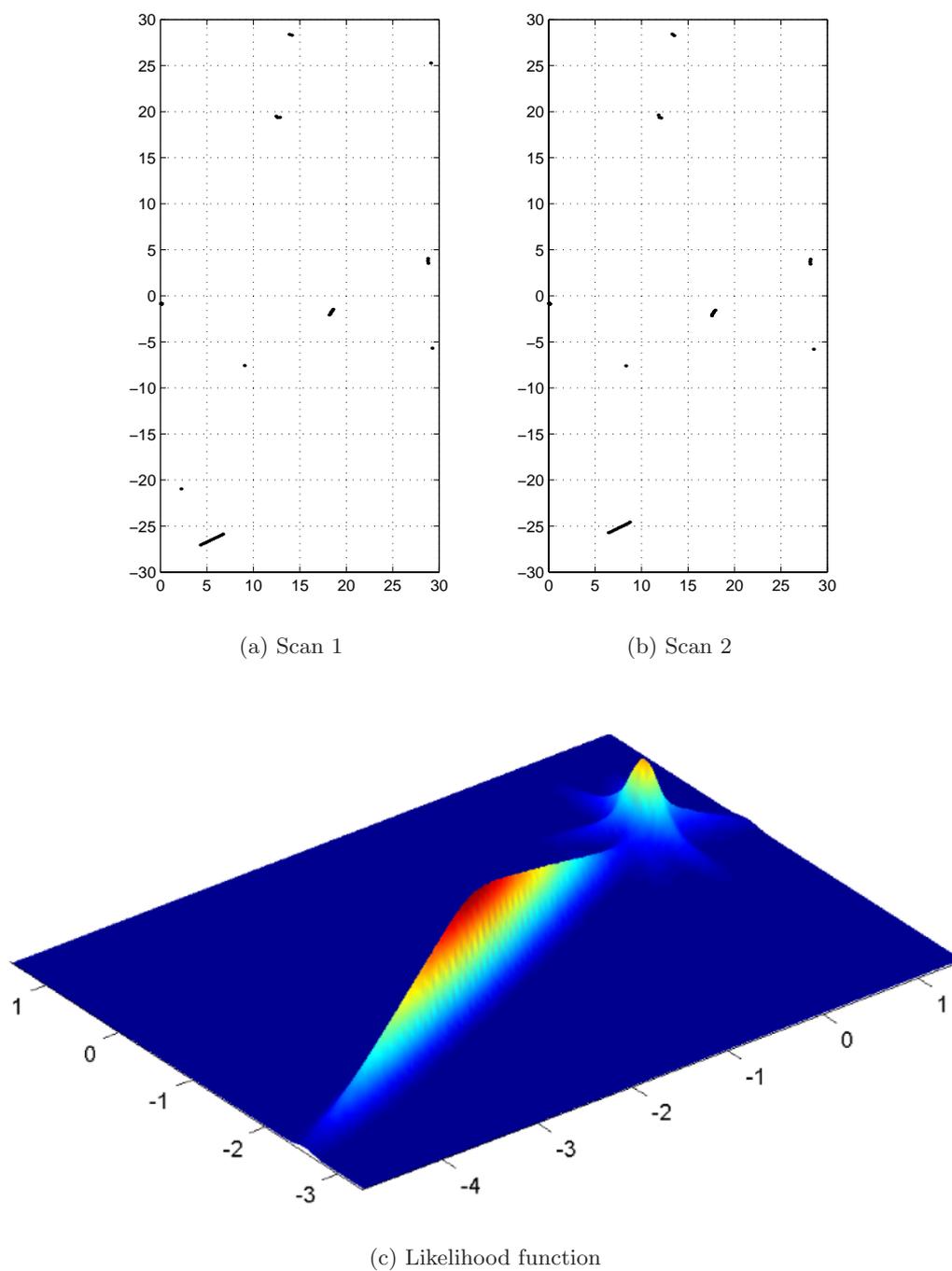


Figure 4.12: Multimodal likelihood function due to dynamic object (motor vehicle).

4.5.3 Discussion of Results

One of the key points to be learned from these results is that unprocessed scan correlation, which uses all the available information, is not necessarily more accurate than feature-based methods that use only part of the available information. This counter-intuitive outcome may be primarily attributed to two factors. First, the ability of feature models to reject spurious data makes use of *a priori* knowledge that certain formations in the data are more reliable than others. For example, data that resembles a tree trunk is likely to be reliable, while disordered data, such as a ground sweep, is not. Second, explicit data association, when correct, is a significant source of information and markedly constrains the resulting relative pose estimate (as shown in Section 4.4.6).

In environments where reliable feature extraction is difficult, scan correlation is undeniably a valuable approach. The results in the mining tunnel demonstrate the quality of the alignment accuracy possible from distinctive, but difficult to model, data.

The multimodal results from the park environment show that dynamic objects (e.g., people, cars, ground sweeps) can induce strong alternative modes, which can cause maximum-likelihood scan correlation to fail. Thus, in environments susceptible to multiple modes, it may be necessary to maintain a full representation of the vehicle pose PDF.

4.6 Application: Particle Filter Localisation

This section presents an implementation of particle filter localisation in the mining tunnel environment.⁸ An off-line map of the mine is created from the laser data and dead reckoning results of the previous section. A different set of laser data, where the vehicle travels back along the tunnel in the opposite direction, is used as the observation data set.

The off-line map was created in a rather *ad hoc* manner to produce a single reference Gaussian sum. Basically, the unprocessed laser points are located in the global reference frame according to the maximum likelihood dead reckoning estimate. Each point represents a unit-height Gaussian, as before, and this set of Gaussians is culled—retaining points on the basis of minimum determinant, representing minimal spatial area—to reduce the number of points while maintaining a reasonable density. In other words, points are compared with other points in a small neighbourhood, and the point with the smallest determinant is kept, while the others are removed from the map. Thus, the map Gaussian sum is composed of a minimal representative point set.

This experiment uses a naive realisation of the particle filter—without provision for the problems of sample impoverishment or inadequate sample coverage. It is assumed that the process noise provides sufficient roughening to prevent impoverishment, and the sensor noise is broad enough to permit reasonable overlap between the prior and posterior distributions.

4.6.1 State Vector and Process Model

The vehicle state is represented by its position and velocity as follows.

$$\mathbf{x}_k = [x_k, y_k, \phi_k, \dot{x}_k, \dot{y}_k, \dot{\phi}_k]^T$$

⁸Discussion of particle filter concepts and implementation is provided in Appendix D.

Note, the pose (x_k, y_k, ϕ_k) is the vehicle pose with respect to the *global* coordinate frame, while the velocity $(\dot{x}_k, \dot{y}_k, \dot{\phi}_k)$ is the change-in-pose with respect to the *vehicle-centred* coordinate frame. This form permits a straightforward realisation of a constant velocity process model.

The six-element state means that a constant velocity state transition model can be derived that satisfies Markov assumptions (i.e., the predicted state $\hat{\mathbf{x}}_{k+1}^-$ is dependent only on the previous state estimate $\hat{\mathbf{x}}_k^+$). The process model, therefore, is given by

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_k + \Delta T(\dot{x}_{k+1} \cos \phi_k - \dot{y}_{k+1} \sin \phi_k) \\ y_k + \Delta T(\dot{x}_{k+1} \sin \phi_k + \dot{y}_{k+1} \cos \phi_k) \\ \phi_k + \Delta T \dot{\phi}_{k+1} \\ \dot{x}_k + \Delta T q_1 \\ \dot{y}_k + \Delta T q_2 \\ \dot{\phi}_k + \Delta T q_3 \end{bmatrix}$$

where ΔT represents the change-in-time, and q_1 , q_2 and q_3 are independent random variables representing the uncertain change in velocity due to accelerations. The values for q_1 , q_2 and q_3 are drawn from zero-mean Gaussian distributions with standard deviations $1.5m/s^2$, $1.5m/s^2$, and $0.3rad/s^2$, respectively. Notice that the process model uses the predicted velocities for time $k+1$ to predict the vehicle pose. These values introduce the acceleration uncertainties into the pose prediction and, in doing so, add roughening noise to the samples of the next prior; thus, uncertainty in pose is due entirely to acceleration uncertainty.

4.6.2 Likelihood Modification

The structure of the mine environment necessitates a modification to the likelihood function obtained directly from scan correlation. The problem is that the robustness of cross-correlation to outlier measurements produces an over-conservative likelihood in the direction of the mine walls.

A typical likelihood function from the mine environment is shown in Figure 4.13, which illustrates the extended likelihood tails in the direction parallel to the mine walls. (Note, the likelihood is reasonably narrow in the perpendicular direction.) Essentially, the characteristic wall texture, which indicates the most likely vehicle pose, is not very distinctive for the scan correlation method described in this chapter (since it fails to penalise outlier measurements). The alternative scan correlation method in Section 4.4.5 might perhaps address this problem without having to resort to the *ad hoc* modification described below.

The solution chosen in this application is to use the basic cross-correlation method (i.e., where the observation forms a single PDF for correlation with the reference PDF) and modify the resulting likelihood to reduce the tail thickness. The modification is simply to subtract half the maximum sampled likelihood from the likelihood weight of each pose sample.⁹

$$\Lambda_k(i) = \Lambda_k(i) - \frac{\max\{\Lambda_k(i)\}}{2}$$

Likelihood samples with $\Lambda_k(i) < 0$ are assigned zero likelihood $\Lambda_k(i) = 0$.

⁹The likelihood of each state sample is dependent only on the pose portion (x_k, y_k, ϕ_k) of the state vector, not the velocity portion $(\dot{x}_k, \dot{y}_k, \dot{\phi}_k)$.

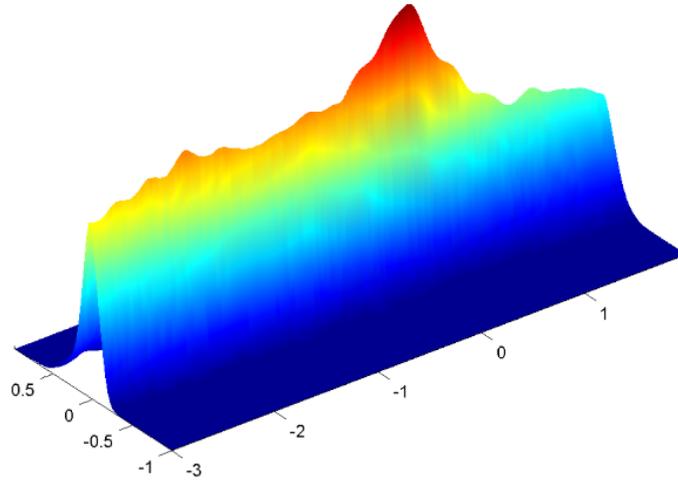


Figure 4.13: Likelihood function along a straight tunnel section. Cross-correlation produces reasonably narrow peak in the direction perpendicular to the walls but is over-conservative along their course.

4.6.3 Results

An initial sample set of 3000 particles is drawn such that the pose estimate has mean $[81m, 96m, -2rad]$ with uniform distribution in the range $[\pm 6, \pm 8, \pm 3.5]$, and the velocity estimate has mean $[3.9m/s, 0.0046m/s, 0.0046rad/s]$ with Gaussian distribution governed by the standard deviations $[\cdot 2, \cdot 2, \cdot 02]$. The “true” initial state was somewhere close to $\mathbf{x}_0 = [84.8, 100.9, -2.1, 3.8, 0.0036, 0.0055]^T$.

With each subsequent iteration of the particle filter, the number of resampled particles is 1000. The first six iterations are shown in Figure 4.14. Notice that after the first iteration, the sample distribution is multimodal, and this quickly settles into two dominant modes due to symmetries in the map structure. The distribution then converges to the true mode after six iterations. Notice also that this implementation does not make use of “coastline” information, which would constrain particles to within the mine walls (i.e., prior samples outside the mine walls would have zero likelihood); this constraint is not particularly advantageous here after initial convergence.

The localisation results for the entire run are shown in Figure 4.15. This figure gives the mean pose estimate for each iteration as a black line and, at selected intervals, shows the full prior and posterior sample sets. Note, the apparent discontinuity at coordinates (35, 25) is actually a 3-point turn.

The (x, y) posterior standard deviation reached as high as $1.5m$ in the direction parallel to the tunnel walls along some straight sections. However, it was typically less than $0.1m$ in the direction perpendicular to the walls, and similarly less than $0.1m$ in all directions as the vehicle approached distinctive regions such as tunnel junctions. Clearly, this particle-based localisation result is quite accurate and robust.

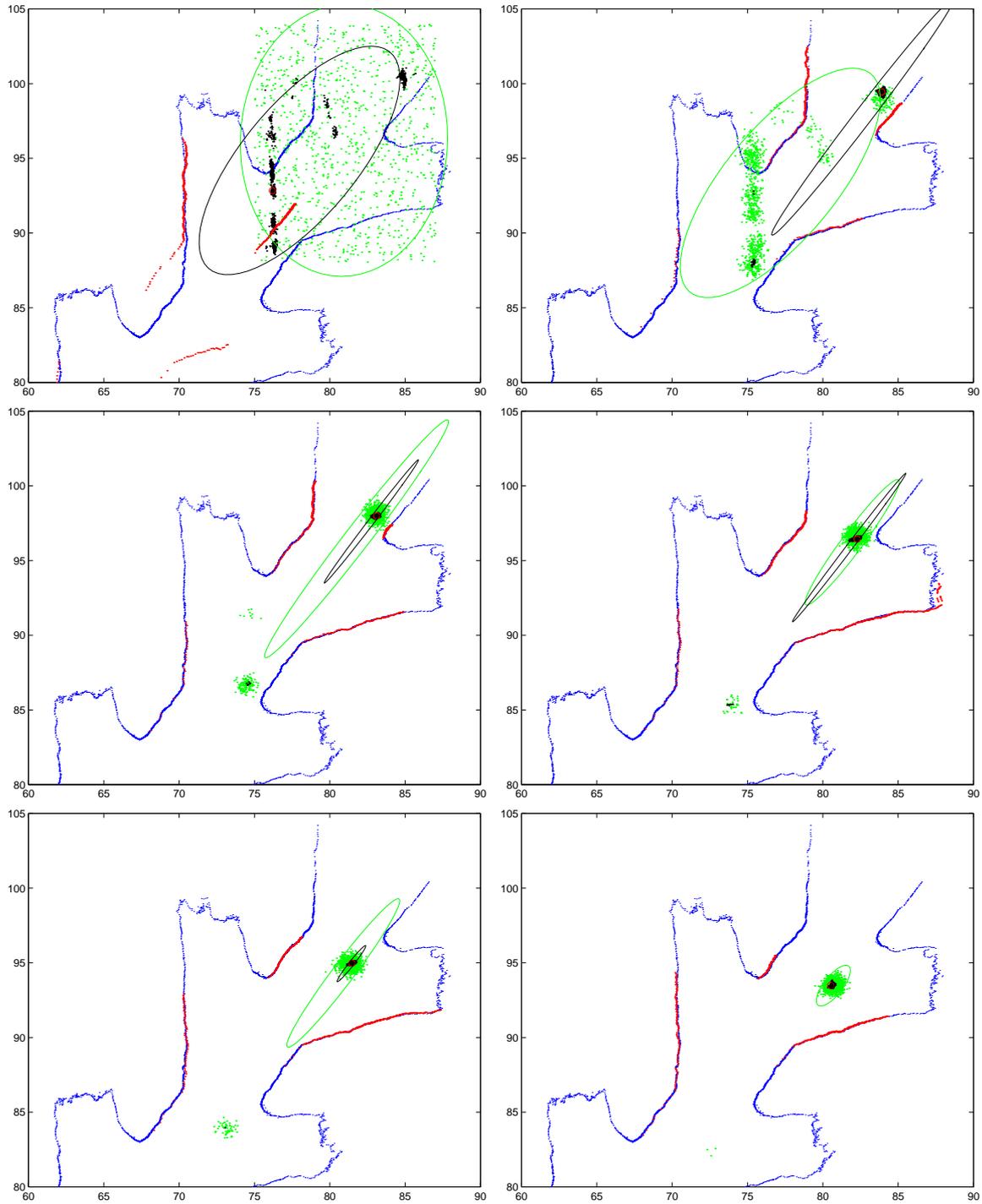


Figure 4.14: First six iterations of localisation. The green samples are the prior (with green sample covariance ellipse). The black samples are the posterior (with black sample covariance ellipse). The red points are the unprocessed laser data projected from the most likely posterior sample.

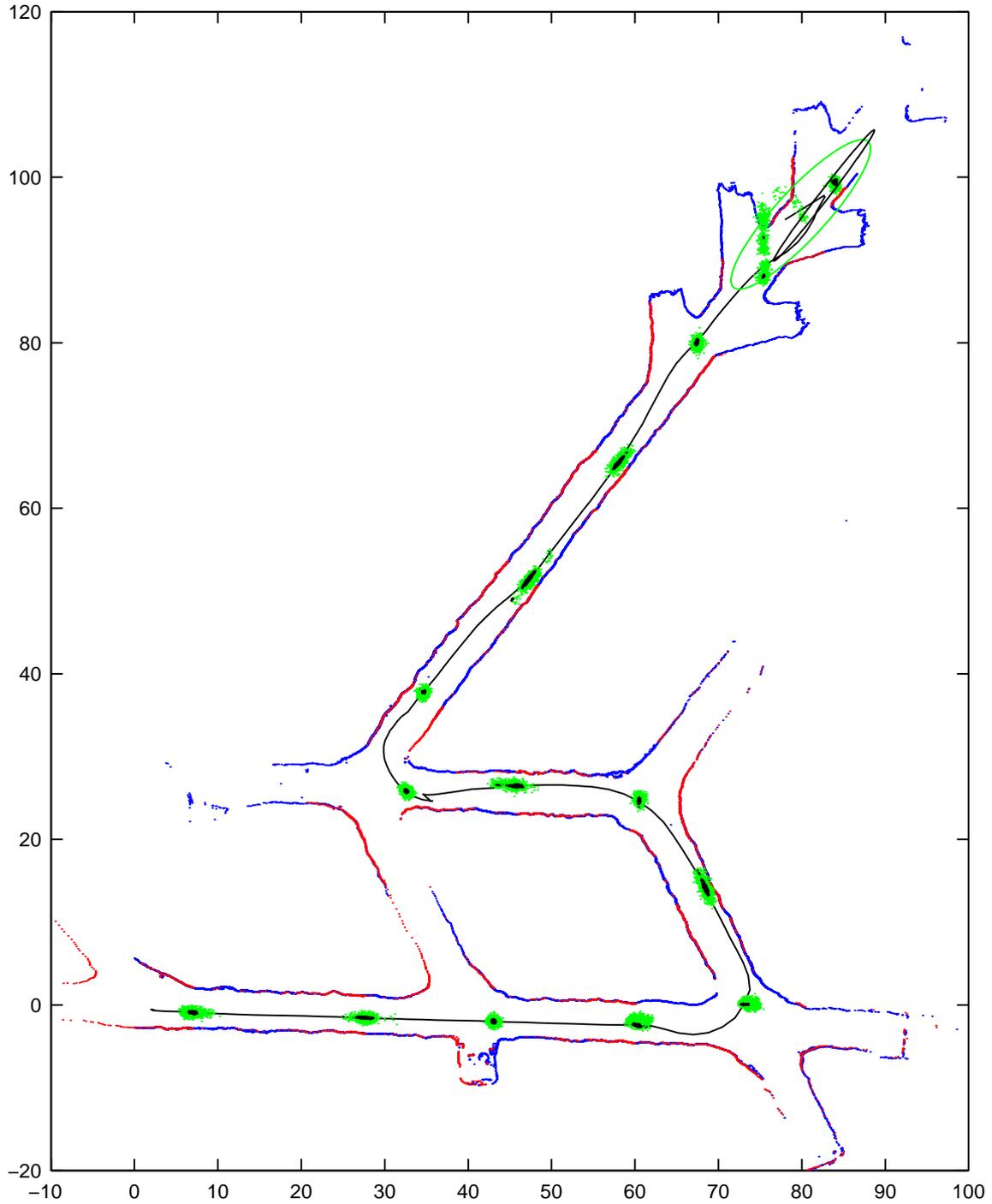


Figure 4.15: Particle filter localisation. The mean pose for each iteration is depicted by the black line. The prior (green) and posterior (black) samples are shown at periodic intervals along with the unprocessed laser scan (red).

4.7 Remarks: A 2-D PDF for Map Building and SLAM

This chapter is, in part, a response to the work in [95] with regard to the use of a 2-D Cartesian parameter-space as the probabilistic representation of a 2-D environment. A main goal of this chapter is to show when and where a 2-D PDF is an appropriate representation, and how it should be interpreted.

The work in [95] presents experimentation in a subsea environment using a scanning sonar sensor. Each radiating sonar “ping” is modelled by a sum-of-Gaussians uncertainty in range and a Gaussian in bearing, indicating the likelihood of a point-target location. This polar PDF is then transformed to observer-centred Cartesian space. The individual ping PDFs, for a full rotation of the sonar, are accumulated into a joint “scan” PDF via multiplication.¹⁰ The scan PDFs may then be used to perform localisation (through scan correlation), map building (multiplication of the scan PDF with an accumulating map PDF, given location), or SLAM.

The fundamental assumption in [95] is that a 2-D PDF is capable of representing the shape of a complex environment and can provide the basis for multiple sensor Bayesian data fusion.

This chapter contends that the use of a 2-D PDF for map building or SLAM is flawed; map building will produce nonsense and SLAM will diverge. The Gaussian sum representation for a sonar ping is valid, but its fusion with other pings via multiplication is incorrect.

Inherently, the 2-D PDF represents the uncertainty distribution of a single point-location target. Thus, the sampled amplitudes of a sonar ping, which measure the likelihood of a point target versus range, are reasonably represented. The Gaussian sum approximation of the ping is also reasonable, and the set of Gaussians in the sum might be interpreted as saying “the true state is here *or* here.” On the other hand, the fusion of measurements by multiplication is equivalent to saying “the true state is here *and* here.” This is not a reasonable interpretation of the data, as each ping does not observe the same point target but different aspects of a non-point surface. Recursive estimation with this data in a 2-D parameter-space is inconsistent.

The apparently reasonable results in [95] are most likely an artifact of using Gaussians (because of their infinite tails), and a counter example can be given by considering a ping with bounded distribution in bearing. In this case, map building (for example) would fail since multiplication of the map PDF with the ping PDF would set every part of the map outside the bearing bounds irrevocably to zero likelihood. Circumventing this problem, by specifying which portions of the map are affected, would involve extremely *ad hoc* tinkering with the estimation process.

The conclusion of this chapter is to claim that a 2-D PDF representation is suited only for tasks like scan correlation and localisation. At most, it might be used to align a set of scans, using a batch process (possibly EM), such that the resulting “map” is the sum of the aligned distributions, which might be used for localisation. It is not suitable for recursive estimation problems, such as map building or SLAM, which will inevitably converge (collapse) to a single point.

¹⁰Accumulation into a “scan” PDF is just an implementation detail to simplify scan correlation. Localisation, map building and SLAM might equally be performed using the individual ping PDFs directly.

4.8 Summary

This chapter examines scan correlation as an alternative to feature-based data association. Scan correlation involves finding the relative pose between two unprocessed point data sets, so that the data sets are aligned. A review of current correlation methods—ICP, angle histogram, occupancy grids, and probabilistic methods—argues that most are deficient in their use of sensor uncertainty. The goal of this chapter, therefore, is to formulate a scan correlation method that incorporates sensor uncertainty appropriately.

If a scan of data is represented in Cartesian space by a 2-dimensional PDF (or 3-D PDF for a 3-D data set), this distribution is implicitly interpreted as defining a single point target. Higher-dimensional parameter-spaces might permit more realistic interpretations of the data, but are subject to other limitations (e.g., they require feature models, are computationally intractable, or lack theoretical justification).

A Bayesian likelihood function for scan correlation, using the “point-location target” model, is shown to be computed via cross-correlation of the two scan PDFs.

A 2-D sum of Gaussians is proposed as a PDF representation for point data sets where the individual point distributions are Gaussian. This representation is shown to permit efficient scan cross-correlation. Two variants of scan correlation are presented. The first is straightforward cross-correlation between two scan PDFs, and the second involves cross-correlation of the reference scan PDF with individual measurement Gaussians from the observation scan. The first variant is robust to outliers and viewpoint variation, while the second is less robust but can be more accurate.

Gaussian sum scan correlation is demonstrated via maximum likelihood dead reckoning in an outdoor environment and in a subterranean mining tunnel. This application gives good results but maximum likelihood estimation is fragile for multimodal likelihood functions. A second application for scan correlation is particle filter localisation, again in the mine tunnel, which shows that the likelihood function gives an accurate and conservative estimate of pose uncertainty.

Chapter 5

Considerations for SLAM in Moderate-Sized Environments

This chapter is concerned with the practical realisation of stochastic SLAM in small to moderate scale environments. The qualifier “moderate” refers to an environment size where the number of stored features (i.e., computation) and non-linearities are small enough to permit straightforward implementation of the SLAM algorithm as presented in Section 2.2.

The first consideration, particularly in dynamic environments with non-static and temporarily static entities, is feature management—the addition and deletion of map features. Without mechanisms to minimise the addition of non-static or unstable features, and to remove obsolete or redundant features, the SLAM map will progressively accumulate clutter. Eventually, this clutter will lead to either excessive computational load or data association failure. Thus, feature management is an essential requirement for successful long-term deployment.

The second consideration concerns the cycle detection problem. While batch data association (such as CCDA) yields reliable association within local regions, it is an insufficient mechanism for robust cycle detection, and may provide little better reliability than individual (i.e., one assignment at a time) association. This chapter describes when this problem will occur and a simple method for its solution.

The feature management and cycle detection concepts are illustrated with experimental SLAM results, performed using the basic SLAM implementation described in Section 2.2 and the CCDA algorithm.

5.1 Feature Management

Feature management is a major contributing factor to SLAM being a harder problem than *a priori* map localisation; not only is it essential to ensure correct associations between observed and stored features, but it is also necessary to manage measurements that fail these validation tests. Non-matched observations might be due to dynamic objects, anomalous feature extraction, outlier measurements of map landmarks, or measurements of landmarks previously unseen. SLAM requires that the last of these cases is detectable, while rejecting the others, so as to permit map extension.

This section is concerned with three aspects of feature management for dynamic environments.

- Feature addition. Non-static or unstable features are avoided by deferring addition into the map until satisfactory evidence of stability is obtained. The method of *constrained initialisation* is examined as a mathematically consistent way to incorporate tentative observation information.
- Removal of redundant features. It is not necessary to retain all static features to build an effective map, and the removal of extraneous features (i.e., density control) can deliver substantial computational saving. Measures are discussed for assessing feature reliability and information content as deletion criteria.
- Removal of obsolete features. If the structure of the environment changes over time, those landmarks ceasing to exist should be removed from the map. By deleting features that are predicted visible, but are not observed, the map automatically adjusts to change.

5.1.1 Feature Addition

The standard procedure for deferring feature addition is to add new features to a “tentative” list, and transfer features from this list to the map once they have been reobserved sufficiently often. Thus, on obtaining a new observation, an attempt is made first to associate it with a feature in the map, then to a feature in the tentative list and, failing that, it becomes a new tentative feature. A series of associations to a particular tentative feature relegates it to the “confirmed” map, while failure to reobserve a tentative feature results in its demise.

In [53], SLAM is performed using sonar, which is “notorious for exhibiting drop-outs, false returns, no-returns and noise.” To circumvent this problem, all measurements not matched to known map features are stored in a list as potential features. These are retained for N time-steps and are discarded if not reobserved during this time. Each time-step a search for clusters¹ of $M \leq N$ measurements is made in the tentative list and, if found, each cluster defines a new confirmed map feature.

A similar approach is presented in [42], where each tentative landmark has a counter for the number of times it is reobserved. A tentative feature is transferred to the map once a sufficient number of associations have occurred, or deleted if not observed for a given period of time.

5.1.2 Constrained Initialisation

The issue of feature initialisation is, in essence, an issue of data association—whether or not a set of subsequent observations correspond to a particular tentative feature. In ascribing

¹Clustering is determined via pair-wise compatibility using the NIS validation gate. Each measurement in a cluster must be from a different time-step and must be compatible with every other measurement in the cluster. Typically, the NIS calculation for tentative features is not performed correctly; this is addressed in Section 5.1.2.

associations, however, the calculation of appropriate validation gates for tentative features is a non-trivial task and is frequently done incorrectly. A related problem is the matter of transferring the accumulated observation information to the map once a feature has been confirmed, and this is usually carried out in a very suboptimal manner. In the discussion below, the method of *constrained initialisation* [140] is shown to address both of these problems.

The exposition in [42] provides a good example of the usual approach to the initialisation problem. A potential feature \mathbf{x}_i is obtained from an observation $\mathbf{z} = [r, \theta]^T$ with covariance \mathbf{R} .

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \mathbf{g}_i(\mathbf{x}_v, \mathbf{z}) = \begin{bmatrix} x_v + r \cos(\theta + \phi_v) \\ y_v + r \sin(\theta + \phi_v) \end{bmatrix} \quad (5.1)$$

The covariance of this feature is given by

$$\mathbf{P}_i = \nabla \mathbf{g}_{\mathbf{x}_v} \mathbf{P}_v \nabla \mathbf{g}_{\mathbf{x}_v}^T + \nabla \mathbf{g}_{\mathbf{z}} \mathbf{R} \nabla \mathbf{g}_{\mathbf{z}}^T \quad (5.2)$$

where $\nabla \mathbf{g}_{\mathbf{x}_v}$ and $\nabla \mathbf{g}_{\mathbf{z}}$ are the Jacobians of $\mathbf{g}_i(\mathbf{x}_v, \mathbf{z})$ with respect to \mathbf{x}_v and \mathbf{z} , respectively, as given in Equations 2.29 and 2.30 in Section 2.2.4 (notice that Equation 5.2 is simply the bottom diagonal of Equation 2.31). Having similarly obtained another potential feature \mathbf{x}_j , the innovation (and innovation covariance) between them is calculated as follows.

$$\nu_{ij} = \begin{bmatrix} \hat{x}_i - \hat{x}_j \\ \hat{y}_i - \hat{y}_j \end{bmatrix} \quad (5.3)$$

$$S_{ij} = \mathbf{P}_i + \mathbf{P}_j \quad (5.4)$$

Association validation is then determined via the NIS gate.

$$M_{ij} = \nu_{ij}^T S_{ij}^{-1} \nu_{ij} < \gamma_2 \quad (5.5)$$

There are two suboptimal consequences of this approach. First, the validation gate is under-constrained since the innovation covariance S_{ij} does not include the correlations between features \mathbf{x}_i and \mathbf{x}_j (resulting from their correlations with the vehicle). Thus, associations may be made that should not be. Second, since these correlations have been lost, only the final observation can be used when initialising a confirmed feature into the map; all preceding observation information for the feature will be lost.

Constrained initialisation [140] is an optimal and consistent mechanism for feature initialisation, which allows immediate addition of tentative features into the map but defers data association until the feature is confirmed. This method permits fully constrained data association and conserves all tentative observation information.

Consider the following example where there is an augmented state vector $\mathbf{x}_a = [\mathbf{x}_v^T, \mathbf{x}_m^T]^T$ that is subsequently augmented with two tentative features \mathbf{x}_i and \mathbf{x}_j .² The covariance of the map may, therefore, be represented by

$$\mathbf{P}_a = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{vm} & \mathbf{P}_{vi} & \mathbf{P}_{vj} \\ \mathbf{P}_{vm}^T & \mathbf{P}_m & \mathbf{P}_{mi} & \mathbf{P}_{mj} \\ \mathbf{P}_{vi}^T & \mathbf{P}_{mi}^T & \mathbf{P}_i & \mathbf{P}_{ij} \\ \mathbf{P}_{vj}^T & \mathbf{P}_{mj}^T & \mathbf{P}_{ij}^T & \mathbf{P}_j \end{bmatrix} \quad (5.6)$$

²State augmentation is performed using the procedure shown in Section 2.2.4. This method is known to be consistent—it adds no information to the older portions of the map and does not involve data association.

If, at some stage, the features \mathbf{x}_i and \mathbf{x}_j (and, perhaps others) are found to represent the same (new) landmark, they can be combined by constraining their estimates. On the other hand, if a tentative feature is not reobserved within a reasonable period, it is deleted from the map.³

Data association is performed by way of a perfect “virtual” observation $\mathbf{z}_{ij} = \mathbf{x}_i - \mathbf{x}_j = \mathbf{0}$, stating that the two features are equivalent.

$$\hat{\mathbf{z}}_{ij} = \mathbf{h}_{ij}(\hat{\mathbf{x}}_a) = \begin{bmatrix} \hat{x}_i - \hat{x}_j \\ \hat{y}_i - \hat{y}_j \end{bmatrix} \quad (5.7)$$

This constraint has the following innovation and innovation covariance.

$$\nu_{ij} = \mathbf{z}_{ij} - \mathbf{h}_{ij}(\hat{\mathbf{x}}_a) = -\mathbf{h}_{ij}(\hat{\mathbf{x}}_a) \quad (5.8)$$

$$S_{ij} = \nabla \mathbf{h}_{\mathbf{x}_a} \mathbf{P}_a \nabla \mathbf{h}_{\mathbf{x}_a}^T \quad (5.9)$$

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_a}$ is given by

$$\nabla \mathbf{h}_{\mathbf{x}_a} = \left. \frac{\partial \mathbf{h}_{ij}}{\partial \mathbf{x}_a} \right|_{\hat{\mathbf{x}}_a} = \begin{bmatrix} 0 & \dots & 0 & 1 & 0 & -1 & 0 \\ 0 & \dots & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \quad (5.10)$$

From the sparsity in $\nabla \mathbf{h}_{\mathbf{x}_a}$, a more efficient evaluation of the innovation covariance is possible.

$$S_{ij} = \nabla \mathbf{h}_r \begin{bmatrix} \mathbf{P}_i & \mathbf{P}_{ij} \\ \mathbf{P}_{ij}^T & \mathbf{P}_j \end{bmatrix} \nabla \mathbf{h}_r^T \quad (5.11)$$

where the reduced Jacobian $\nabla \mathbf{h}_r$ is

$$\nabla \mathbf{h}_r = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \quad (5.12)$$

The resulting NIS value (using Equation 5.5) correctly includes the cross correlations \mathbf{P}_{ij} for properly constrained gating.

Having confirmed that a set of tentative features represent the same landmark, a final constrained estimate is obtained through the normal Kalman filter update equations presented in Section 2.2.3 (using Equations 5.8 and 5.9 in place of Equations 2.17 and 2.18). This causes the constrained features to each possess the same value (and correlations), and so the set of redundant estimates can be simply deleted from the map. A complete discussion of constrained initialisation is provided in [140], including proof that the solution is equivalent to that obtained without deferred data association.

5.1.3 Density Control

In many environments, it is possible to obtain many more static landmarks than are necessary for accurate navigation, and the increase in information is outweighed by the increase in computation and storage. This is particularly so when features are unevenly distributed

³Deletion of a feature from the SLAM map has been shown to be consistent [40] and, in the case of feature initialisation, no information is lost save the location of the tentative feature.

or clustered. The aim of density control, therefore, is to determine which subset of features to retain, and which to remove, so as to obtain a “best” minimal map.

Presented in [40] are two important characteristics of the stochastic SLAM algorithm with regard to feature removal.

- A feature can be deleted (by removing the appropriate elements from the state vector, and the associated rows and columns of the state covariance matrix) without compromising the statistical consistency of the map.
- The accumulated information of a deleted feature must be discarded, and cannot be reinstated if the feature is reobserved; the feature must be initialised using only the new observation information. Conversely, reusing the old information will result in an inconsistent map.

In [40], the information content of a particular feature \mathbf{x}_i is specified as the *inverse* of the *trace* of its covariance \mathbf{P}_i (a submatrix of the state covariance). The following strategy is proposed to permit feature removal with minimal information loss. At each time-step, those map features transitioning from visible to not-visible are stored in a list and, whenever the vehicle has travelled a distance interval greater than d_v , a single feature from the list is retained (i.e., the one with maximum information) and the rest are discarded from the map. The result is that the landmark density outside the vehicle field-of-view becomes closely related to the distance d_v . Using this strategy, experimental results in an indoor environment showed that over half the map features could be deleted without significant change in SLAM accuracy.

Another criterion for feature removal is a measure of “quality” as suggested in [42], where at regular intervals (perhaps again using the distance d_v) those features below a fixed quality threshold are deleted. The quality measure given in [42] is determined from the innovation sequence for each feature as follows. Suppose feature \mathbf{x}_i has been observed n times, then the innovation sequence for this feature is $\{\nu_1, \dots, \nu_n\}$ with covariance $\{S_1, \dots, S_n\}$. The quality of this feature, therefore, is given by

$$Q_i = \frac{\sum_{j=1}^n \frac{1}{2\pi\sqrt{|S_j|}} \exp\left(-\frac{1}{2}\nu_j^T S_j^{-1} \nu_j\right)}{\sum_{j=1}^n \frac{1}{2\pi\sqrt{|S_j|}}} \quad (5.13)$$

which is the normalised sum of association likelihoods (see Equation 3.2 in Section 3.1.2).⁴ This quality measure is essentially an indicator of feature *stability* and may also be useful as a criterion for feature addition.

In this thesis, a third criterion for feature removal is proposed: feature *visibility*. This expresses that the utility of a particular feature is dependent on the size of the region from which it is observable. In a sense, feature visibility is similar to information content—the more frequently a feature is observed, the more certain its location—however, there is a significant difference, which is well illustrated by the case where the vehicle is stationary. Repeated observations of a feature from a fixed location serves to ever reduce its uncertainty

⁴Note, the likelihood formulation in Equation 5.13 represents the particular case where the innovation vector is of dimension 2 (i.e., for a range-bearing observation of a point feature).

(i.e., increase information) but says nothing about the size of its visibility region, which may be arbitrarily small due to occlusion from surrounding objects.

For the SLAM problem, feature visibility is generally more important than information content, but this is not necessarily always the case. The essential measure of a feature’s utility is the frequency with which it is observed—thereby enabling the robot to gauge its pose—and observation frequency is a function of the size of the feature’s visibility region and the quantity of time the robot spends in that region. As a measure of utility, the size of the visibility region indicates *potential* observation frequency, and is the better measure if the robot trajectory is unconstrained (i.e., the robot may travel anywhere within the environment). On the other hand, information content provides an indication of *past* observation frequency, and is therefore a better utility measure if the robot is constrained to follow previous actions and trajectories (e.g., the robot trajectory might consistently favour a less visible feature). Note, further complication to measuring utility is introduced by the interaction of multiple features. For example, a feature with small visibility region and low information content becomes important if it is the only feature visible from a certain pose. Another feature may be of better quality, but might always share visibility with other near features.

The following metric for quantifying visibility is used in this thesis. A feature’s visibility upon initialisation is zero and, whenever the feature is observed for two consecutive scans, its visibility rating is increased by the distance travelled by the vehicle over that interval. While this metric is not a very accurate representation of visibility (and, no doubt, better metrics are possible), it was found to be sufficient for the experimental purposes of this thesis. Essentially, it is a compromise between visibility and information content; it does not actually specify a region of visibility, but rewards feature utility as a function of both observation frequency and distance travelled.

The strategy for feature removal, therefore, is as follows. The vehicle field-of-view is specified by a bounded region and, each time-step, a set of features transition from inside to outside this region. If a transitioning feature has no neighbours closer than d_v , it is retained. If, however, it has a neighbour closer than d_v with greater visibility, it is removed. On the other hand, if the neighbour has lower visibility and is also outside the field-of-view region, the neighbour is removed.⁵ A demonstration of this strategy in practice is shown in Figure 5.1. Note that the field-of-view region may be arbitrarily greater than the actual sensor field-of-view and, in this example, extends 10 metres behind the vehicle.

5.1.4 Obsolete Feature Removal

Landmarks become obsolete if they move, are removed, or become permanently occluded. For example, people or vehicles might remain stationary for arbitrary periods of time, and be considered good quality landmarks by the SLAM algorithm. In the longer term, structural changes may occur in the environment—furniture shifted, trees or poles removed. An object

⁵A problem with this strategy can arise if a highly reliable feature becomes permanently occluded for some reason. Other near features may be unable to accumulate sufficient visibility to remove it; they are always deleted first. One way to circumvent this problem is to maintain the visibility count of all valid landmarks—even after removal. (This approach assumes that correct association can be made to a deleted feature once it is reinitialised.)

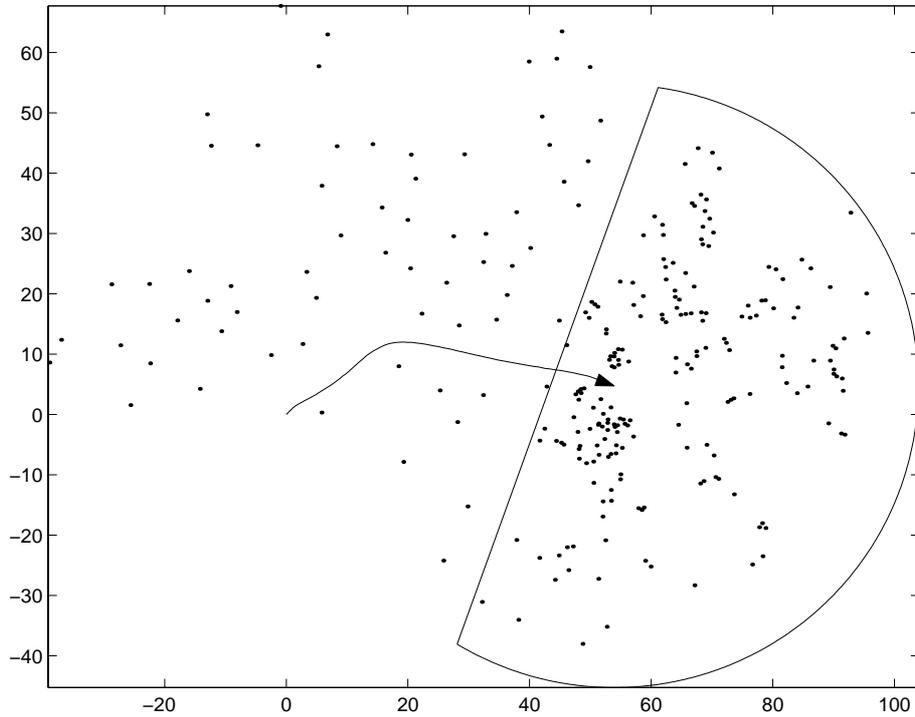


Figure 5.1: Density control. The vehicle performs SLAM along the trajectory shown (using a laser-sensor in an outdoor environment) with the final location depicted by the triangle. As map features move out of the prescribed field-of-view region they are culled to obtain a minimum spacing of $d_v = 4$ metres.

might be placed in front of a landmark, occluding it from view. For whatever reason, some landmarks may cease to exist and no longer provide useful information. These obsolete features should be deleted from the map to maintain an uncluttered and contemporary representation of the environment.

The density control methods in Section 5.1.3 tend to remove obsolete features that are in close proximity to good features. This discussion focuses on removing obsolete features that may not be near other objects.

In [53], if a map feature is predicted to be visible, but is not observed over several consecutive time-steps, it is deleted. A probability of detection $P_D < 1$ is assumed such that the probability of a feature’s existence, if not observed over n time-steps, is $(1 - P_D)^n$. Thus, setting a threshold on n is equivalent to setting a minimum probability threshold on the feature’s existence at the predicted location. The explanation in [53], however, does not specify criteria for “predicted visibility.” For example, it does not mention whether the vehicle uncertainty or models of occlusion are taken into consideration.⁶ These details are the subject of the method presented below.

⁶Another visibility consideration, not addressed in this thesis, is the range of vehicle headings (i.e., angular range) over which a feature is predicted visible. This is important for features such as edges (i.e., range discontinuities) that might only be visible from particular directions.

In this thesis, obsolete features are detected by projecting map features into sensor-space and comparing their expected location with the current observation set. The approach described is specific to a scanning laser sensor, where the set of measurements can be connected to form a polygon, such that map features are either inside (i.e., predicted visible) or outside (i.e., occluded or outside the field-of-view).

One implementation is to convert each map feature \mathbf{x}_i to an observer-centred Cartesian frame as follows.

$$\hat{\mathbf{z}}_{C_i} = \mathbf{h}_i(\hat{\mathbf{x}}_a) = \begin{bmatrix} (\hat{x}_i - \hat{x}_v) \cos \hat{\phi}_v + (\hat{y}_i - \hat{y}_v) \sin \hat{\phi}_v \\ (\hat{y}_i - \hat{y}_v) \cos \hat{\phi}_v - (\hat{x}_i - \hat{x}_v) \sin \hat{\phi}_v \end{bmatrix} \quad (5.14)$$

$$\mathbf{R}_{C_i} = \nabla \mathbf{h}_{\mathbf{x}_a} \mathbf{P}_a \nabla \mathbf{h}_{\mathbf{x}_a}^T \quad (5.15)$$

where the Jacobian

$$\nabla \mathbf{h}_{\mathbf{x}_a} = \left. \frac{\partial \mathbf{h}_i}{\partial \mathbf{x}_a} \right|_{\hat{\mathbf{x}}_a}$$

is obtained in the same way as shown in Equations 3.20 to 3.25 of Section 3.3.1. The resulting estimate will lie either inside or outside the scan polygon as shown in Figure 5.2(a). If the feature is sufficiently inside the polygon, (e.g., by 2σ), then it is marked as “obsolete” (i.e., predicted visible and not observed). This might be accomplished practically by sampling equispaced points about the feature’s 2σ ellipse, and subjecting them to a point-in-polygon test [70]. Note, in the figure, the sensor uncertainty is assumed much smaller than the feature uncertainty, and is ignored. However, measurement uncertainty might be included by shortening each range measurement by $2\sigma_r$.

Since the laser scan is a set of angle-ordered range-bearing measurements, a much more efficient implementation of this test is possible. The map features are projected into polar sensor-space as follows.

$$\hat{\mathbf{z}}_i = \mathbf{h}_i(\hat{\mathbf{x}}_a) = \begin{bmatrix} \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_i - \hat{y}_v}{\hat{x}_i - \hat{x}_v}\right) - \hat{\phi}_v \end{bmatrix} \quad (5.16)$$

$$\mathbf{R}_i = \nabla \mathbf{h}_{\mathbf{x}_a} \mathbf{P}_a \nabla \mathbf{h}_{\mathbf{x}_a}^T \quad (5.17)$$

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_a}$ is given in Equation 2.20 of Section 2.2.3. The 2σ ellipse of $\hat{\mathbf{z}}_i$ is then tested for being inside the (θ, r) polygon as shown in Figure 5.2(b). This may be performed by sampling the ellipse as before and checking whether, for each sampled value (θ_s, r_s) , the two theta values in the laser scan bounding θ_s possess values for r greater than r_s .

5.2 Application: Partial SLAM

This application demonstrates an interesting type of feature management, where *all* features outside the vehicle field-of-view are removed. The field-of-view, in this case, is specified as a (40m radius) circular region about the vehicle. This application is termed *partial SLAM* since it represents an intermediate form between *full SLAM*, where all features are stored, and the sensor-based dead reckoning of Section 3.4, where only the features from the previous scan are retained.

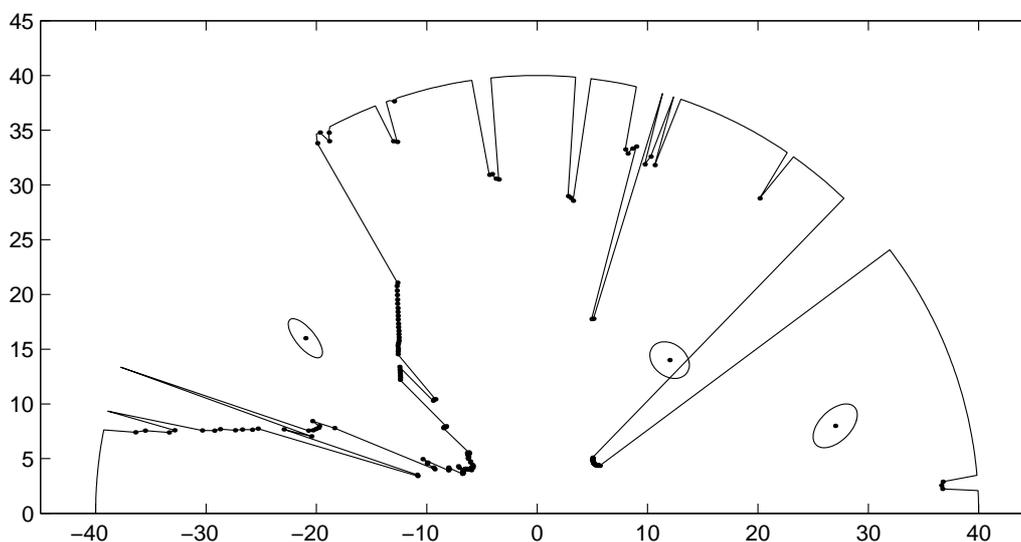
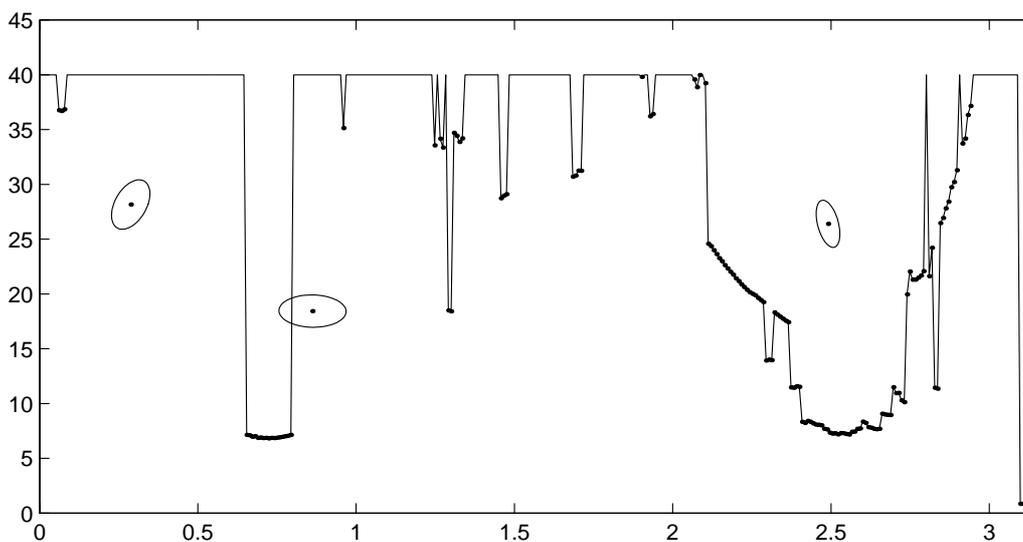
(a) Cartesian (x, y) (b) Polar (θ, r)

Figure 5.2: Obsolete feature removal. The 180° unprocessed laser scan defines a polygon relative to the sensor location. The range of these measurements is limited by a “no return” maximum (40 metres in this case). Features from the SLAM map are transformed to the observer coordinate space and compared with the polygon bounds. If the 2σ ellipse of a feature is inside the polygon it is marked as obsolete. Figures (a) and (b) represent the same information, (a) in observer-centred Cartesian space, and (b) in polar space. In Figure (a), the feature on the right is removed and the other two are retained.

Sensor-based dead reckoning, through alignment of consecutive scans, enables (almost) temporally uncorrelated motion estimation, but produces pose estimates with monotonically increasing uncertainty over time—even if the vehicle is standing still. Partial SLAM, on the other hand, is temporally correlated, and produces pose estimates that converge with time, but diverge slowly with spatial motion.

Partial SLAM is identical to full SLAM for the case where the vehicle is motionless or moves only in one direction (or, more precisely, does not return to old regions of the environment). While full SLAM experiences convergence when retracing old regions, partial SLAM suffers unbounded uncertainty growth with all translational motion beyond its current field-of-view. This growth tends to be very slow, however, and has a corresponding benefit of bounded computation due to the bounded map size. Also the problems with cycle detection (see Section 5.3 below) tend not to be an issue since the map size is of the same order as the sensor range, and is strongly correlated.

5.2.1 Results

Experimental results of partial SLAM are shown in Figure 5.3, where the feature extraction and other implementation details are the same as for sensor-based dead reckoning in Section 3.4. While the accuracy of this result appears similar to the results in Section 3.4, there is a significant difference in the estimate of uncertainty, as shown in Figure 5.4.

The covariance of the sensor-based dead reckoning result (in Section 3.4) tended to grow very quickly due to over-conservative tuning of the measurement uncertainty and, particularly, occasional loss of tracking (i.e., if insufficient features were matched between scans, the sensor data was ignored and the change-in-pose estimate was simply the very uncertain prediction). Because of the gross effects of occasional tracking loss on the rapid growth in pose uncertainty, these graphs were not shown in Section 3.4; they were essentially meaningless in the long-term.

However, for partial SLAM, with the same tuning parameters, quite reasonable estimates of uncertainty were obtained as shown in Figure 5.4. This was because storage of local features, and their reregistration, served to permit temporal convergence of the pose estimate, in spite of conservative tuning of the measurement uncertainty. The most significant improvement was the ability to reregister after temporary tracking failure, such that the very great increases in uncertainty from being lost over short-term scan sequences could be resolved. For example, scans 1743 to 1760 failed to register, resulting in extreme pose uncertainty, but this was reduced subsequently with the registration of scan 1761.

5.2.2 Partial SLAM and GPS

Partial SLAM on its own is not particularly useful; it essentially provides a highly accurate, albeit temporally correlated, dead reckoning estimate. It has potential value, though, as an auxiliary information source in conjunction with an observer of absolute pose, such as GPS. While GPS may be intermittent, noisy, and inaccurate (depending on satellite availability), partial SLAM acts both as a low pass filter on the absolute measurements and an interpolating estimate between GPS signals. The GPS information can be incorporated into the SLAM equations using a trivial observation model such as $\mathbf{z}_{gps} = \mathbf{x}_v$ with uncertainty \mathbf{R}_{gps} .

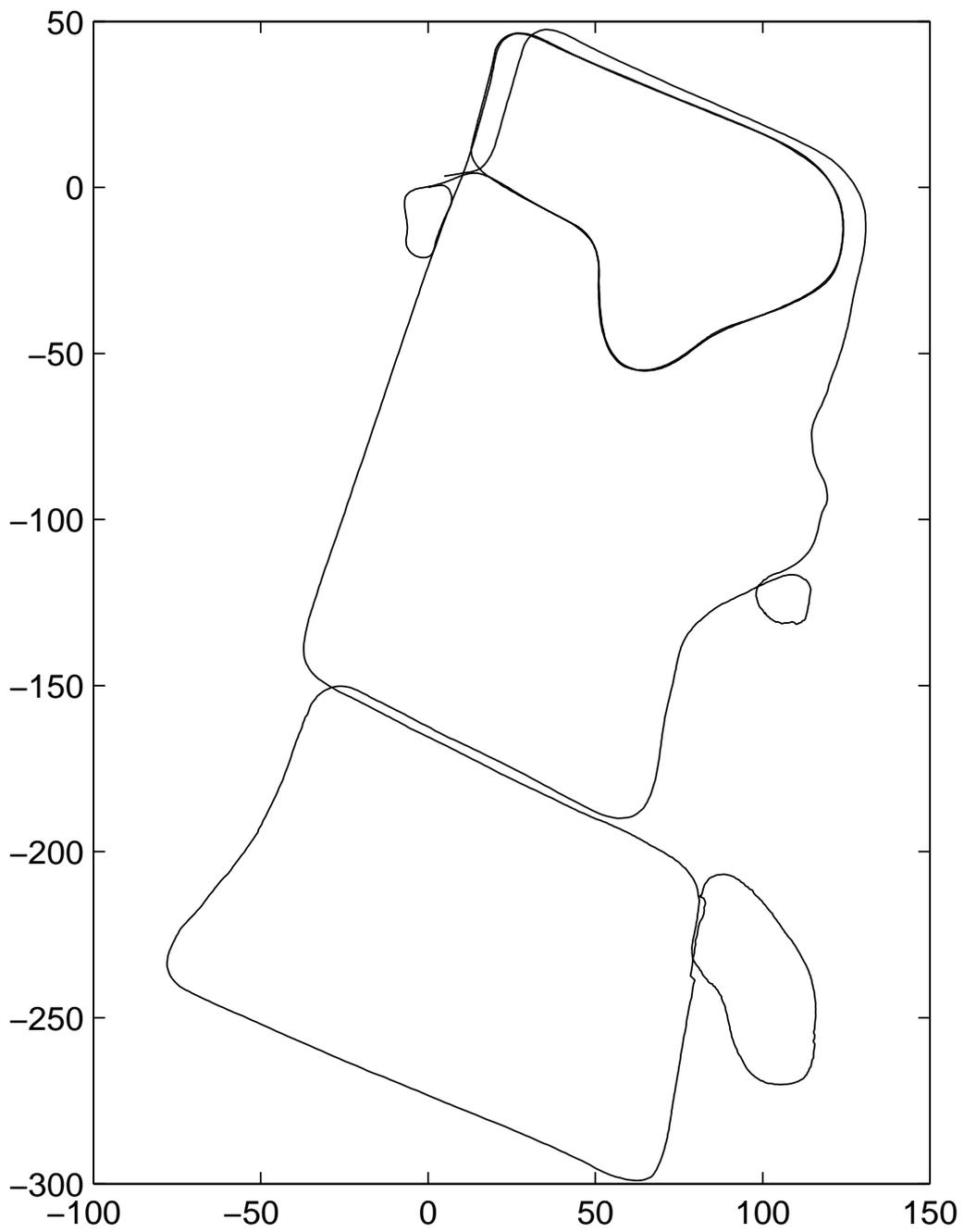


Figure 5.3: Partial SLAM in the internal road environment.

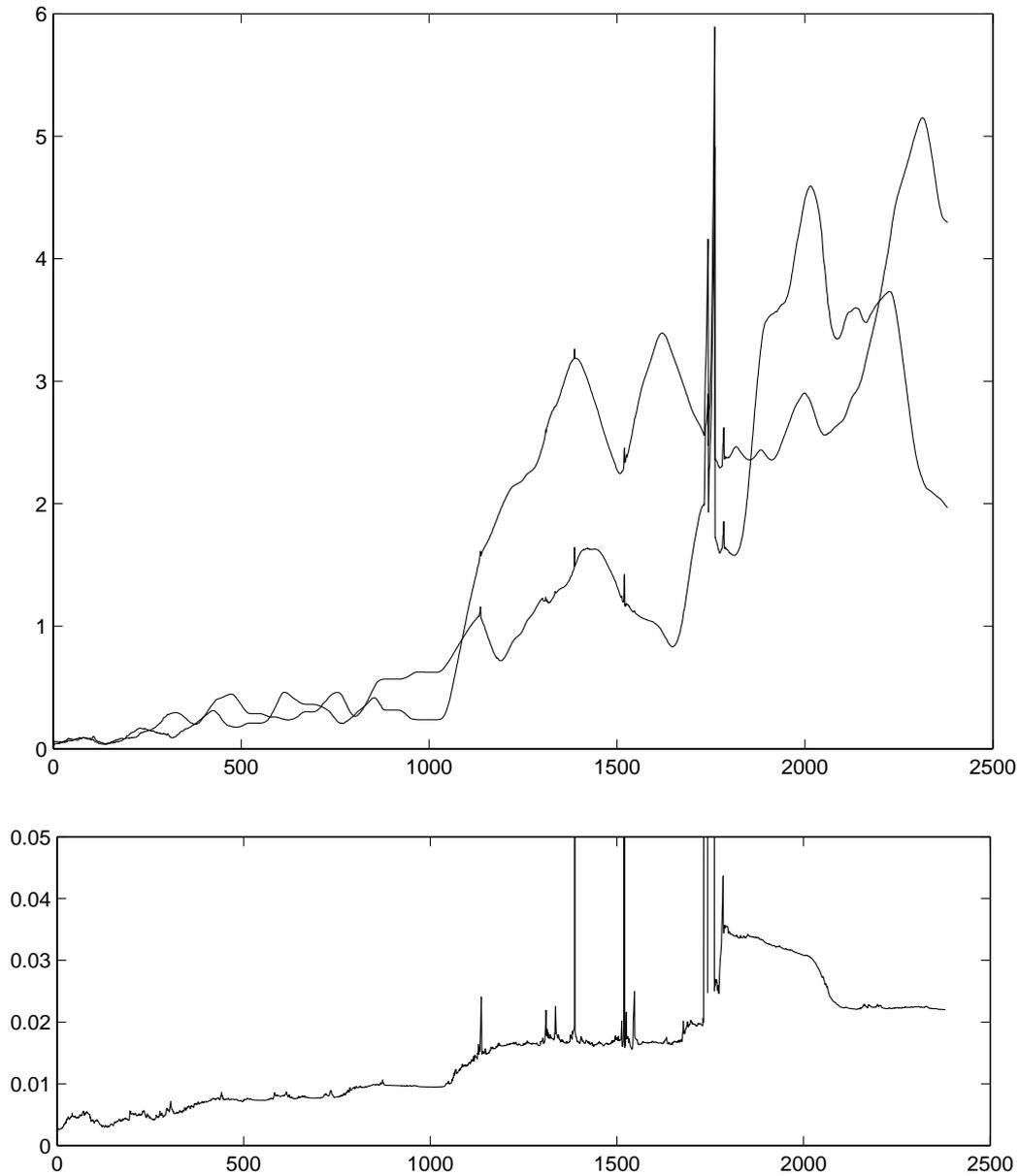


Figure 5.4: Partial SLAM covariance estimates. The top figure shows the x_v and y_v standard deviations (scan number versus metres), and the bottom shows the ϕ_v standard deviation (truncated at 0.05 radians; peaked 0.5 at scan 1760). Notice that the uncertainty ever increases with vehicle motion because feature information outside the vehicle field-of-view is lost. However, sudden large increases in uncertainty, due to temporary loss of tracking, are recovered by subsequent reregistration with the local map.

5.3 Data Association and the Cycle Detection Problem

Batch data association permits robust association for *a priori* map localisation and for most situations in SLAM. However, the cycle detection problem introduces a difficulty not directly addressed by batch association methods. This difficulty arises from the weakness of correlation between new and old parts of the map when closing the loop, which implies weak relative constraints and reduced benefit in using a batch process.

Consider the example in Figure 5.5. The vehicle returns to an old map region via a large loop, all the while creating new map landmarks. Since the vehicle has not yet reregistered with the old map features, its pose uncertainty relative to those features is very large. Furthermore, the uncertainty of the new map features with respect to the old map is also large (i.e., large covariance diagonals and weak cross-covariance terms). Suppose the vehicle makes an observation scan where it observes several of the new map features and another feature not previously observed (perhaps spurious), but the latter observation is (wrongly) associated to an old map feature. The batch constraints in the association process do not prevent this catastrophic failure because the relative constraints between the new and old map features are so weak—the association to the old map feature may just as well have been performed individually. As a result of this association, the map irretrievably diverges since the relative location of the new map portion is updated, and hence constrained, incorrectly.

In general, the problem occurs if (i) there exist groups of features in the map that are strongly correlated to each other but very weakly correlated to features in other groups, and (ii) the batch association assigns a disproportionate number of observations to some groups and only one or two to others. Essentially, the reliability of batch association (due to relative constraints) is dependent on the number of associations within each group.

A solution to the cycle detection problem is quite simple. First, group the map into sets of mutually constrained features and, second, perform batch association *separately* for each group. If the same observation is assigned in two different groups, then the participating map features are said to be the same (i.e., constrained as a single feature).⁷ In this thesis, grouping map features was performed using a simple, though only approximately correct, algorithm as follows (see Figure 5.6). Initially there are no groups. For each feature \mathbf{x}_i in the map, if there exists a group where each feature in the group is suitably constrained with \mathbf{x}_i , then \mathbf{x}_i is added to that group. Otherwise, \mathbf{x}_i starts a new group. The metric for “suitably constrained” is a threshold on the covariance of the relative constraint between two features (see Equations 3.15 to 3.19 in Section 3.3.1).

A limitation of this solution to cycle detection is that it lacks checks for false batch associations to entire groups due to the existence of symmetries in the environment. This is a difficult problem which is addressed more completely in Section 6.3.

5.4 Application: Full SLAM

This application presents a full SLAM experiment that incorporates simple feature management, batch data association and cycle detection via feature grouping. The basic im-

⁷Constraint between two features from different groups is performed according to the constrained initialisation method in Section 5.1.2. The redundant feature is subsequently removed. Note, the two groups involved will then merge to form a single strongly correlated group.

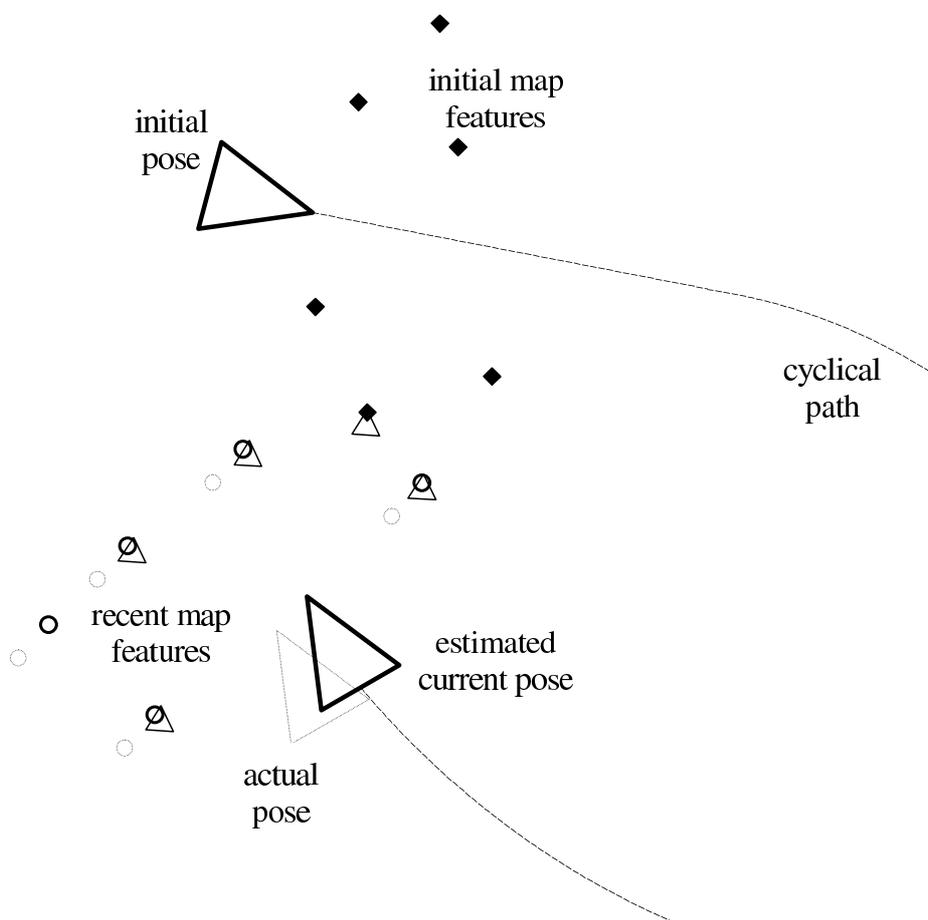


Figure 5.5: The cycle detection problem. Consider, for example, the situation depicted where batch data association fails to prevent a major false association. A vehicle performs SLAM along a cyclic path such that there exists an old map portion (diamonds) and a new map portion (circles, their true locations given by fainter circles). Each of these portions of the map are strongly correlated within themselves but only weakly correlated to each other. The current observation set (triangles) is assigned to the map features using batch data association. Four are assigned to recent map features and one to an old map feature. However, due to lack of geometric constraint, the incorrect assignment to the old map feature is allowed and results in an inconsistent map update. (Note, the falsely assigned observation may have been of a feature not previously seen, perhaps due to occlusion, or simply a false return.)

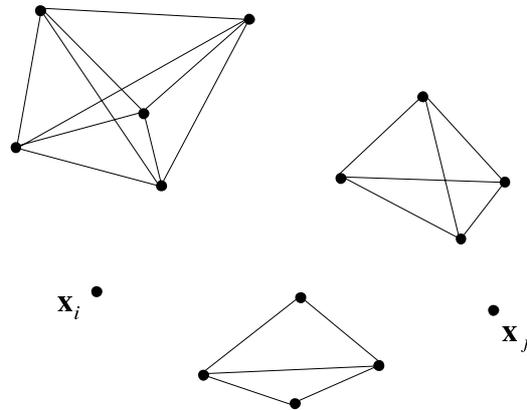


Figure 5.6: Feature grouping. Working sequentially through the vector of map features, groups are formed of features sufficiently united by relative constraints. Each feature \mathbf{x}_i is first checked against existing groups. If a group is found where \mathbf{x}_i is constrained by every feature in the group, then \mathbf{x}_i is added to the group, and the process begins again with the next non-grouped feature \mathbf{x}_j . If, on the other hand, no groups adequately constrain \mathbf{x}_i , then it forms a new group. Once grouping is completed, batch data association is performed separately for each group.

plementation details for this experiment (i.e., feature extraction, inertial prediction model) are the same as described in Section 3.4.

Feature management in this experiment was limited to visibility-based density control only. (A simple *ad hoc* initialisation criterion was also used, where new features were not added to the map if closer than x metres to existing features.) The more sophisticated feature management methods described in this chapter were not implemented, although this test would certainly benefit from constrained initialisation. Furthermore, the SLAM result was not real-time since the naive SLAM algorithm used was insufficient to contend with the large number of features involved.

5.4.1 Results With Medium-Scale Loop Closure

The result of SLAM after the first 1070 scans is shown in Figure 5.7, during which time the vehicle travelled 800 metres. The vehicle first turns clockwise about a small loop, and then twice counter-clockwise around a large 330 metre loop, before heading away along a linear path. The size of the main loop was not large enough to necessitate grouping as the map was strongly correlated throughout (due to the loop size being of similar order to the sensor range and the high feature density in the region). The accuracy of the map is apparent from the closeness of the two loop traversals (on a narrow road), and the structure of the unprocessed laser measurements when projected onto their global locations; they depict precisely the cabin walls and tree trunk shapes.

The SLAM features are shown as circles in Figure 5.7, and can be seen to correspond to objects of near-circular cross-section from the unprocessed laser data (n.b., the circles are iconic and do not represent the estimated radius). These features were culled as they

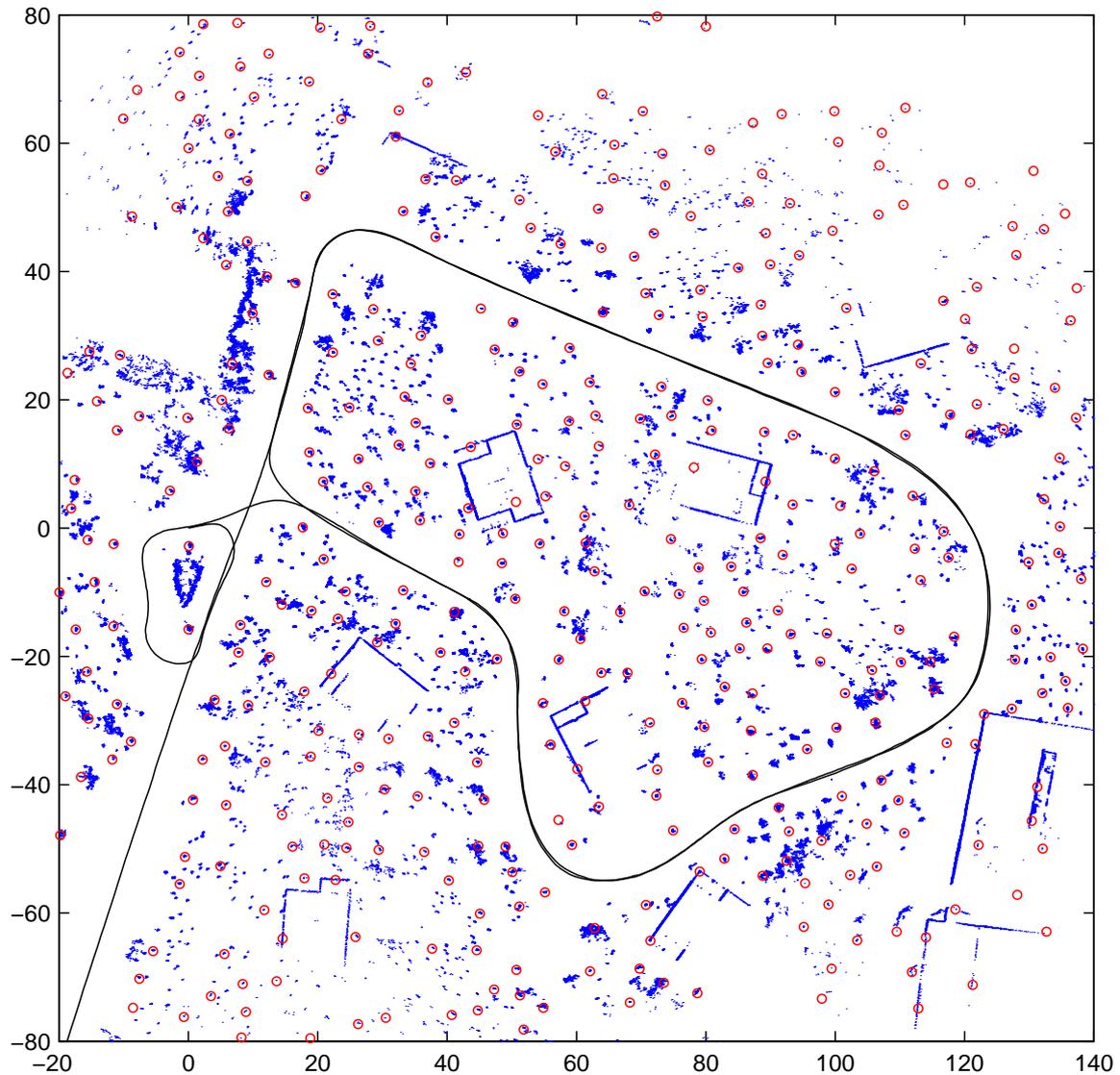


Figure 5.7: Full SLAM in the internal road environment. The path of the vehicle over 1070 laser scans (800 metres) is shown along with the unprocessed laser returns for each pose (depicting shrubs, trees and cabins). The SLAM features are shown as circles. Note, there are two subtle points to be gleaned from this figure. The first is the effect of mild laser tilt, where both front and back edges of the cabin verandas are observed as different laser scans detect slightly different vertical slices of the environment. Second, notice that some features lie on the lines of the cabin walls; these are not spurious, but are poles extending above (or piers below) the cabin floor.

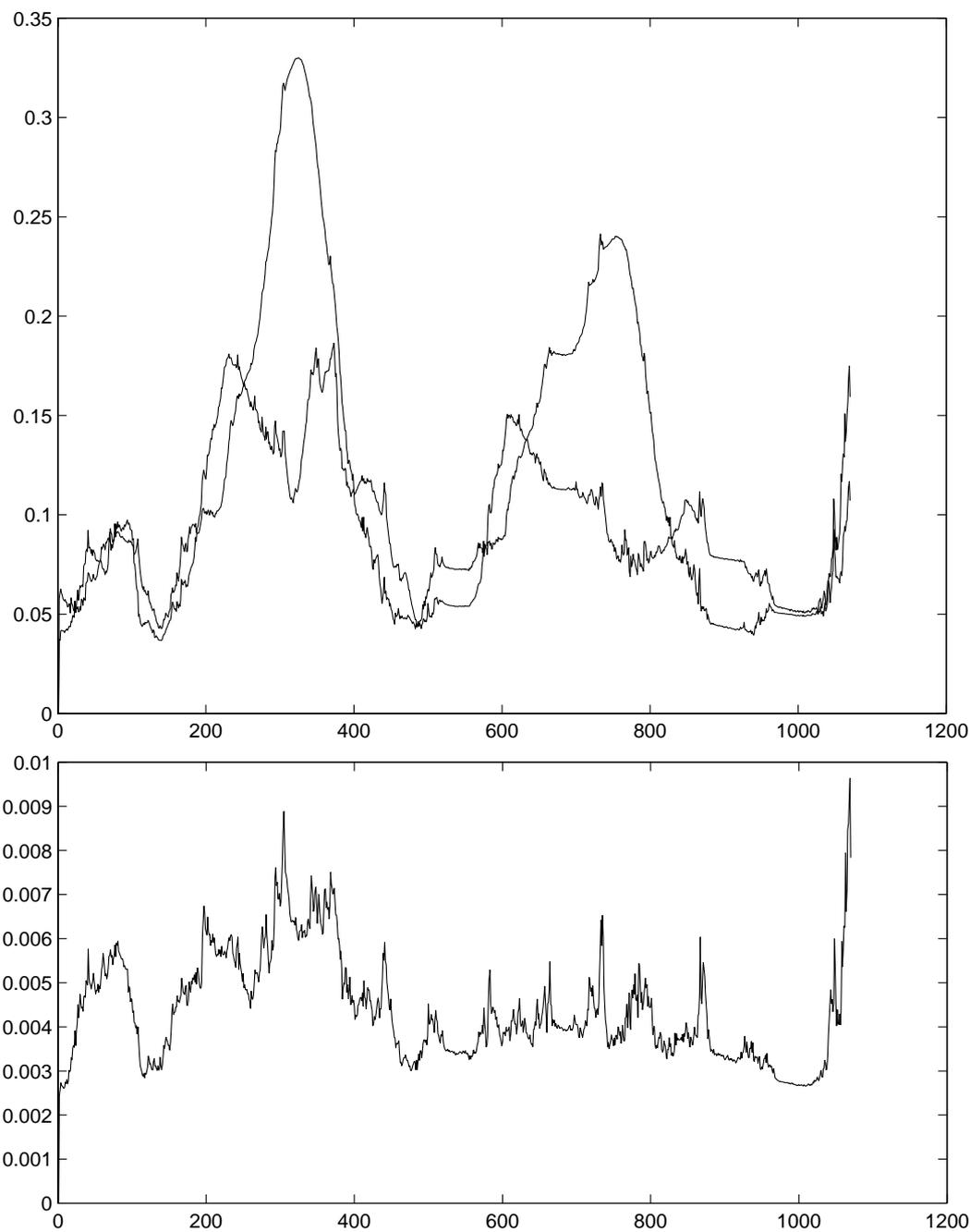


Figure 5.8: Covariance for full SLAM over first 1070 scans (top, standard deviations for x_v and y_v , bottom, standard deviation for ϕ_v). Note, the covariance of the first loop traversal is greater than the second, and returns to a minimum when returning close to the origin (scans 490 and 1000).

moved out of the vehicle field-of-view to obtain a minimum spacing of 4 metres.

The covariance of the result is shown in Figure 5.8. The covariance increases as the vehicle travels away from the origin and reaches a maximum about the extremities of the loop. The covariance falls back to a minimum as the vehicle returns towards the origin. Notice that the map converges with repeated observation, and the peak covariance of the second loop traversal is rather less than the first. The rapid rise in uncertainty at the end occurs as the vehicle again heads away from the origin, this time into an unexplored region with considerably lower feature density.

5.4.2 Results With Large-Scale Loop Closure

The results in Figure 5.9 show the trajectory of the vehicle over 2290 scans, during which it travelled about two kilometres. After completing the dual loop traversal described previously, the vehicle maintained a straight heading until turning left at coordinates $(-35, -145)$, continuing straight until $(80, -200)$ and turning right. At this point, the vehicle performed an off-road counter-clockwise loop and, during this manoeuvre, observed many spurious features (mostly due to a sloping embankment) and few stable features. This resulted in highly uncertain map building, and probably association failures due to the simplistic implementations of the feature initialisation mechanism (no tentative set) and the CCDA algorithm (choose largest clique). Returning to the road, the major clockwise loop was resumed; a region with very few visible features was encountered at the turn at $(-70, -240)$ and, finally, cycle detection and a large error correction was performed on closing the loop. The vehicle then retraced the old map a distance, turning left at $(55, -190)$, making a small clockwise loop at $(100, -120)$, and performing a second major cycle detection (this time with little correction) on rejoining the original map loop.

The two regions mentioned with few useful features tended to break the map into separated regions connected to each other by very weak correlations. Thus, for reliable cycle detection, feature grouping was essential. A detailed view of the main loop closure is shown at the bottom of Figure 5.9. Notice the large error correction and accompanying reduction in pose uncertainty at this point.

The large error correction in this experiment demonstrates a major flaw in the traditional stochastic SLAM formulation. Even with correct data association, the map would definitely become inconsistent after such a large linearised update. In fact, given the prior map uncertainty, it is probable that the map was inconsistent *before* the update. The problem is two-fold: first, the linear (Gaussian) representation of such large uncertainties is likely to be optimistic and, second, the large linearised update (including error propagation throughout the map via correlations) invalidates the basic near-linear, small-error assumptions of the EKF.

Fundamentally, the existence of large uncertainties in the map, so as to necessitate feature grouping for data association, is likely to prevent consistent application of the EKF-based SLAM framework. This problem leads to the development of submap methods, which break SLAM into manageable connected subsets, as presented in the next chapter.

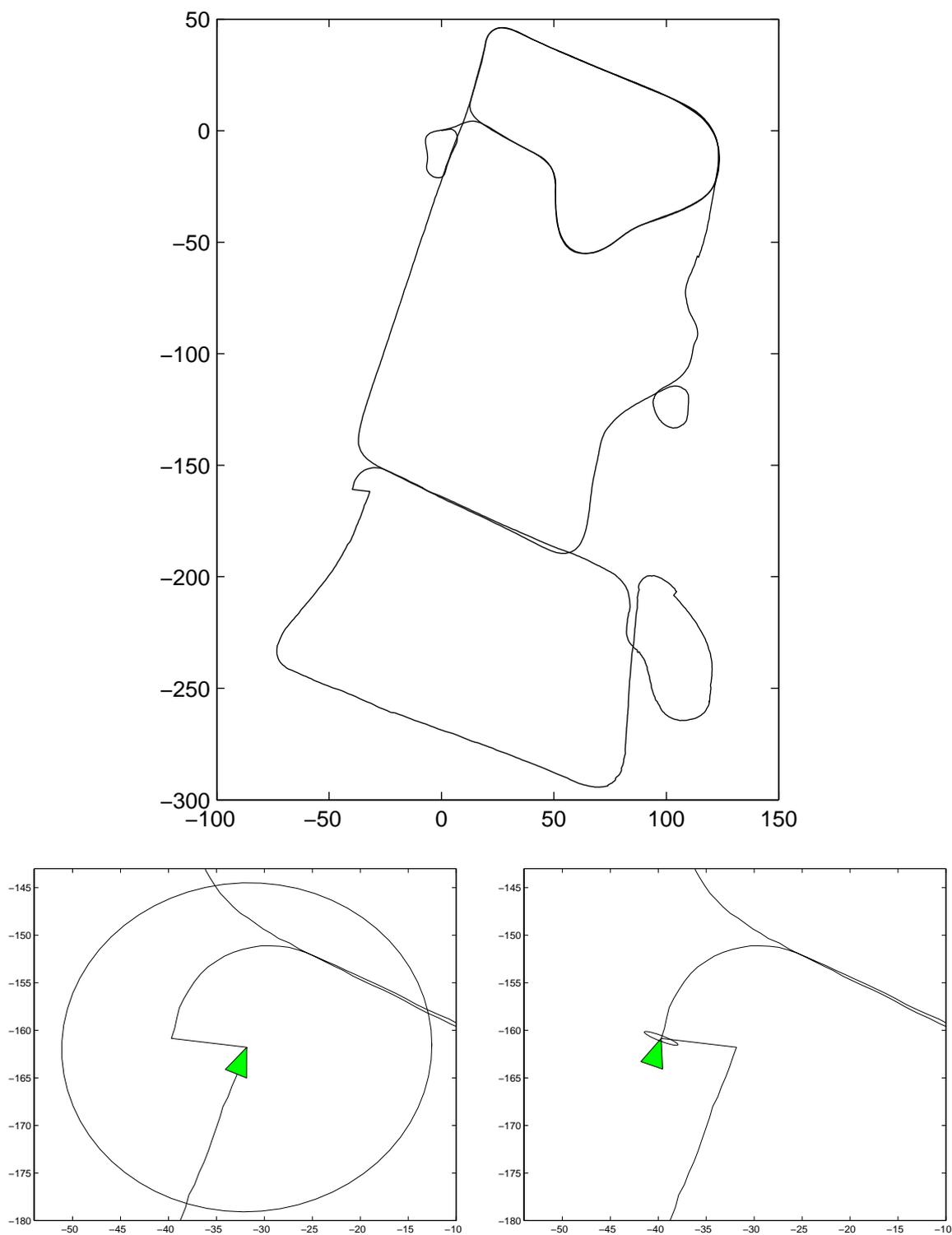


Figure 5.9: Full SLAM with large-scale loop closure. The top figure shows the vehicle path over 2290 scans (2 kilometres), and the bottom figures show detail of the large loop error correction—with the vehicle pose and 2σ covariance ellipse before and after cycle detection.

5.5 Remarks: SLAM versus Batch Map Building

Recent progress in offline batch map building (e.g., using EM [131]) offers a competitive alternative to SLAM, particularly in highly dynamic environments. This section discusses their relative merits and stipulates when an *a priori* batch map might be preferred.

A map produced by SLAM is incremental and requires no prior exploration of the environment; it is used immediately for localisation. It also permits on-line adaptation to structural changes. However, in dynamic environments, SLAM reliability is hindered by the need to consider all objects (static and dynamic) as potential landmarks. In extreme cases, this can lead to data association fragility through map clutter.

Offline map generation using batch procedures has been shown in [131] to permit accurate metric maps with minimal human intervention. These maps can be checked manually (visually) for consistency and are subsequently used as static *a priori* maps. This approach has several marked advantages. First, the map can be used consistently and reliably without a formal uncertainty model (i.e., unlike SLAM, the map does not require a measure of global convergence). The map is formed according to a maximum likelihood criteria, which does not necessarily have to match the true physical shape of the world. Second, if localisation becomes temporarily lost (perhaps through incorrect data association), it can be recovered subsequently by increasing the search space; unlike SLAM, such errors are not self-propagating. Third, measurements of dynamic objects can be identified on-line and filtered from the localisation data set. This technique is demonstrated in [129], where shorter-than-expected range measurements are rejected to permit robust localisation in an environment crowded with people. A disadvantage of offline maps is that the map must be rebuilt whenever there is significant change to the structure of its working environment.

To offer some general rules of thumb regarding the use of SLAM or static batch maps, consider the following criteria. SLAM is best suited if a task requires an immediate estimate of location in a previously unknown environment. It is also suited to environments where the physical structure changes regularly but have only moderate levels of transient objects. In these environments, SLAM can operate robustly given the mechanisms described in this thesis: batch data association serves to extract known landmarks and reject clutter, constrained initialisation prevents the addition of non-static objects to the map, and obsolete feature removal adapts the map to structural change. On the other hand, an *a priori* batch map cannot automatically adjust to environmental change but is very reliable in stable environments with large numbers of transient objects. The static nature of the map means that measurements of transient objects can be easily filtered out, and localisation can be recovered even after a data association failure.

5.6 Summary

This chapter is concerned with feature management and cycle detection for traditional SLAM in medium-sized environments.

Feature management involves addition and removal of map features based on feature reliability and utility. The *constrained initialisation* procedure is reviewed as an optimal method for deferred feature initialisation, which permits delayed data association without

loss of information. Two reasons for feature removal are examined: density control and obsolete features. Density control reduces map clutter and improves efficiency by deleting lesser quality features. Two measures of feature quality are reviewed—information content and innovation sequence stability—and a new measure is proposed: feature visibility, where a feature’s utility is governed by the size of the region over which it is visible. Obsolete feature removal is necessary to delete landmarks that no longer exist (due to environmental change). A feature is marked obsolete if it is predicted visible but not observed and, for this purpose, a laser-specific model of “predicted visibility” is presented.

For large loops, robust cycle detection may require more than just batch data association; rather, it is shown to rely on feature grouping, where the map features are clustered into strongly constrained sets. Batch association must then be performed separately for each set.

Chapter 6

Network Coupled Feature Maps

Stochastic SLAM suffers two major problems in relation to scale: excessive computation and storage costs, and inconsistent (optimistic) uncertainty estimation due to accumulated non-linearities and bias. This chapter proposes that both these problems can be addressed by partitioning the map into a topological-metric network of submaps. Each submap is a small independent stochastic SLAM map, and these are connected by suboptimal, but consistent, coupling estimates.

The strengths of topological and metric maps are complementary. Topological maps provide a natural division of the environment, low computation and storage, large-scale connectivity and consistency, and place recognition through the characteristics of local groupings. Metric maps, on the other hand, provide high local accuracy, quantified uncertainty, generality, and pose constrained data association. Combining these qualities in a hybrid topological-metric map permits feasible consistent SLAM on a very large scale.

This chapter presents a hybrid map framework for stochastic SLAM in extensive environments. The following topics are discussed with regard to this map proposal.

- A review is made of current submap-network systems that use stochastic SLAM as the estimation mechanism for the local submaps. This discussion is primarily concerned with the consistency of the submap coupling methodologies.
- The *network coupled feature maps* (NCFM) framework is introduced. The NCFM concept is presented in terms of the map structure and the methods for performing globally consistent, convergent SLAM.
- The cycle detection problem is examined in detail, with solutions to the issues of very-large-scale search and loop confirmation presented using the NCFM framework. Also discussed is the difficult problem of pathological environmental symmetries, where rejection of false cycles may be practically impossible.
- An application of NCFM SLAM in an outdoor environment is shown to demonstrate the concepts described in this chapter.

6.1 Submap Methods for Stochastic SLAM

This section reviews five recent submap-SLAM proposals, which represent the current state-of-the-art in decoupled stochastic SLAM methods. Each of these methods possess a common theme whereby, at the level of individual submaps, SLAM is performed using (essentially) the traditional EKF equations given in Section 2.2. Furthermore, these submaps are maintained independently so as to avoid correlation with other submaps.

The critical difference between these methods is their approach to estimating the coupling between submaps, and thus connecting the map at a global level. This is the area where submap methods break away from established SLAM concepts and involve new complexities, particularly with regard to the consistent use of information. Therefore, this section is primarily concerned with the consistency of submap coupling estimation and submap transitions, and the submap methods are examined in terms of the following questions.

- Coupling estimation. How are the submaps coupled? Are the coupling estimates consistent (i.e., do they provide a conservative estimate of pose errors at a global level)? Do the coupling estimates converge (i.e., global map convergence) and what are the limits of this convergence?
- Vehicle transition. When the vehicle transitions from one submap to another, how is the pose estimate reinstated in the new submap? What (if any) information is transferred from the previous submap, and is this consistent?
- Loop closure. Is the global structure of the map appropriate for cycle detection and consistent loop closure?

6.1.1 Decoupled Stochastic Mapping

Decoupled stochastic mapping (DSM) [90] is actually a *global* map, but the total region is split into a number of “globally referenced” cells. (In this regard, DSM is distinct from the subsequent submap methods discussed in this section, wherein each submap defines a *local* coordinate origin.) Correlations are maintained between features within each submap, but are neglected between features from other submaps. Thus, the advantage of this method is purely computational; if there were N cells of n features, full SLAM would require $O(N^2n^2)$ computation and storage but DSM reduces computation to $O(n^2)$ and storage to $O(Nn^2)$.

Each submap has its own estimate of the vehicle pose and the locations of features within the cell region—all with respect to the global coordinate frame. All submaps are marked “inactive” except for the single “active” cell currently occupied by the vehicle and, while active, this submap is updated according to the standard SLAM algorithm. The inactive submaps are frozen with their last “active” state estimate (including vehicle pose).

When transitioning from the active submap to another, one of two transition procedures are performed: *cross-map relocation* or *cross-map update*. The former is used when moving into an older submap (where age refers to the order of cell creation), and serves simply to reinitialise the vehicle pose estimate in the newly activated cell. The latter procedure is used when entering a newer submap, and transfers information from the older cell to permit improvement in the global uncertainty of the newly activated cell.

The fundamental contention with DSM is the information transfer between submaps during the “cross-map” procedures. These processes inevitably introduce correlations between the submaps, and the attempts to then decorrelate them using *ad hoc* uncertainty inflation are questionable. In general, the inflation incurs over-conservative submap estimates but, in some circumstances, may yet produce optimistic results (n.b., empirically these procedures appear to yield consistent results). A consistent alternative to *cross-map relocation* is presented in [140], wherein vehicle pose reinitialisation is performed without correlation between the submaps. However, a theoretically consistent alternative to *cross-map update*, necessary for global convergence, is not presently available.

Finally, the globally referenced structure of DSM does not address the problem of loop closure. As with traditional stochastic SLAM, the large-error non-linear update associated with closing a large loop will tend to invalidate the basic EKF assumptions of near-linearity, and result in an inconsistent map estimate.

6.1.2 Two-Level Landmark Representation

In [24] a two-level representational hierarchy is proposed. The first of these is the *landmark level*, where a “landmark” is a set of local features expressed according to a common local coordinate frame. Each landmark also includes an estimate of the vehicle pose relative to its local frame. The second level of representation is the *global level*, where the pose of each landmark (i.e., each landmark coordinate origin) is estimated with respect to a common global coordinate frame. This base frame also has an estimate of the (global) vehicle pose.

The estimation process¹ for each landmark is performed in isolation from every other landmark; similarly, the global frame is also treated in isolation. At each prediction timestep, the vehicle states are propagated independently within each landmark, and also for the global frame. Meanwhile, in the vehicle-centred frame, a temporary landmark is constructed and this is matched against the existing map landmarks. Having obtained a unique landmark match, the features of the map landmark are updated using the geometric constraints of the observation landmark. This same observation information is then, separately, used to update the global estimate of the landmark pose.

The advantage of this two-level approach is that, by ignoring correlations between the landmark and global levels, the computation and storage for SLAM are reduced to $O(N^2 + n^2)$ and $O(N^2 + Nn^2)$, respectively (where N is the number of landmarks and n is the number of features per landmark).

There are three significant problems with this approach. The first is that neglecting the correlations between the two levels (i.e., between the features of a landmark and the global landmark location) must introduce inconsistencies into the global map; although it is possible, in this case, that the level of inconsistency is small and easily subsumed by conservative (inflated) observation uncertainty. The second problem is that the vehicle pose prediction step may introduce very large uncertainties into the local landmark estimates, and these non-linear errors can compromise filter consistency. A solution to this problem is to

¹The underlying estimation process used in [24] is a stochastic SLAM variant called the *symmetries and perturbations map* (SP map) [25]. This approach is reported to provide a uniform representation for information gathered by different sensors, and to overcome the problem of singularities in the representation of geometric features.

remove the vehicle pose elements from the local landmark state while the vehicle is distant, and reinitialise them when it returns to the landmark vicinity (see [140] or Appendix C.2 for details on the vehicle reinitialisation process). The third problem is that, as with DSM above, the structure of the global level does not address the loop closure problem, and the global map estimate may be invalidated by a large non-linear error correction.

6.1.3 Relative Landmark Representation

The *relative landmark representation* (RLR) [64, 66] possesses a similar conceptual structure to the previous method [24]. It represents sets of nearby features in local relative coordinate frames and maintains these local frames according to the global coordinate system. However, unlike the previous method, the RLR manages all the appropriate correlations in a consistent, and near-optimal, manner.

The RLR divides the map into sub-regions where, in each region, the landmarks are represented with respect to a local coordinate frame. The pose of each local frame is defined by a small subset of local features (called “base” features), which are represented in the global coordinate frame. Thus, the SLAM augmented vector consists of: the vehicle states and local frame base features in absolute coordinates, and the remaining local region features in relative coordinates. The vehicle and base features are referred to as “absolute” states and the local features are called “relative” states.

The motivation behind the RLR structure is to reduce correlations between state elements. In traditional SLAM, where vehicle and features are all in global coordinates, the states become increasingly correlated over time. With the RLR, the absolute states become strongly correlated, and the relative states within each local region become strongly correlated to each other and to the absolute states, but the relative states in different local regions are only ever weakly correlated. This is particularly so between relative states from physically distant local frames. The key idea, therefore, is that the relative states in different local regions can be “decorrelated” from each other with little loss of information—resulting in a dramatic reduction in computation and storage requirements.

The method of decorrelation is to inflate the diagonal sub-matrices of the state covariance, so that the respective states can be treated as uncorrelated.

$$\begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C} & \mathbf{B} \end{bmatrix} \xrightarrow{\text{decorrelate}} \begin{bmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{0} & \mathbf{B}' \end{bmatrix}$$

The decorrelated covariance is conservative if the difference between it and the original covariance matrix is positive semi-definite.

$$\begin{bmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{0} & \mathbf{B}' \end{bmatrix} - \begin{bmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C} & \mathbf{B} \end{bmatrix} \geq \mathbf{0}$$

The decorrelation strategy of the RLR is to maintain all strong correlations: absolute-to-absolute, relative-to-absolute, and (within each local region) relative-to-relative; but to decorrelate relative-to-relative states from different local regions. Since the correlations between these non-local relative states are very small, decorrelation can be performed with minor inflation of the covariance diagonals.

Nevertheless, performing decorrelation after each observation would result in a very conservative SLAM estimate, and the RLR achieves its near-optimal convergence properties by employing the *compressed* EKF (CEKF) [65] as its estimation method. The CEKF limits the SLAM update step to only effect the relative and base landmarks in the vehicle’s current neighbourhood (these are termed the “active” states) and postpones the full update, and the decorrelation process, until the vehicle leaves the local region. Thus, decorrelation is performed only intermittently and, when applied, inflates the covariance of the active states but leaves the covariance of the other “passive” states unchanged.

The advantage of the RLR is its $O(Nn^2)$ computation and storage, where N is the number of local regions and n is the number of features per region. Also, it is reported to achieve close to the accuracy and convergence rate of full SLAM (although this depends somewhat on the assumption that the vehicle remains in a local area for a reasonable period of time so as to minimise covariance inflation).

The main issue not addressed by the RLR is non-linearity. The same problems of loop closure and large error corrections affecting traditional SLAM, and the two submap methods discussed previously, will also tend to generate inconsistent estimates with the globally referenced portions of the RLR state vector.

6.1.4 Hierarchical Local Maps

In [29, 30] a hierarchy of local submaps is introduced, where each local submap is maintained independently using traditional SLAM,² and these are connected by estimates of the relative pose between submap origins.

The relative pose between two adjacent submaps is determined at the creation of the second submap as follows. The first submap is built using standard SLAM until the vehicle pose uncertainty reaches a given threshold. Let the augmented SLAM estimate with respect to the first submap coordinate frame S_1 be given by

$${}_{S_1}\hat{\mathbf{x}}_a = [{}_{S_1}\hat{\mathbf{x}}_v^T, {}_{S_1}\hat{\mathbf{x}}_m^T]^T$$

$${}_{S_1}\mathbf{P}_a = \begin{bmatrix} {}_{S_1}\mathbf{P}_v & {}_{S_1}\mathbf{P}_{vm} \\ {}_{S_1}\mathbf{P}_{vm}^T & {}_{S_1}\mathbf{P}_m \end{bmatrix}$$

At this point, the second submap S_2 is created, with the current vehicle pose as its origin. Thus, the pose of the second submap relative to the first is equal to the vehicle pose portion of the first submap state estimate (i.e., ${}_{S_1}\hat{\mathbf{x}}_{S_2} = {}_{S_1}\hat{\mathbf{x}}_v$ and ${}_{S_1}\mathbf{P}_{S_2} = {}_{S_1}\mathbf{P}_v$).

The set of coupling estimates $\{{}_{S_i}\hat{\mathbf{x}}_{S_j}, {}_{S_i}\mathbf{P}_{S_j}\}$, therefore, are extracted from the relevant state estimates and stored in a *coupling tree*, which represents the order of submap creation. Consider, for example, a situation where the vehicle constructs and traverses a set of submaps in the order shown in Figure 6.1(a). The coupling tree for this traversal sequence is shown in Figure 6.1(b). The relative pose between two non-adjacent submap coordinate frames can be calculated from the “monotonic linkage” between the submaps—the vector sum of connecting relative pose estimates in the coupling tree. For example, the relative

²In fact, [29, 30] implement a suboptimal variant of SLAM called *relocation-fusion* [102], and use an *unscented filter* [76] rather than the standard EKF.

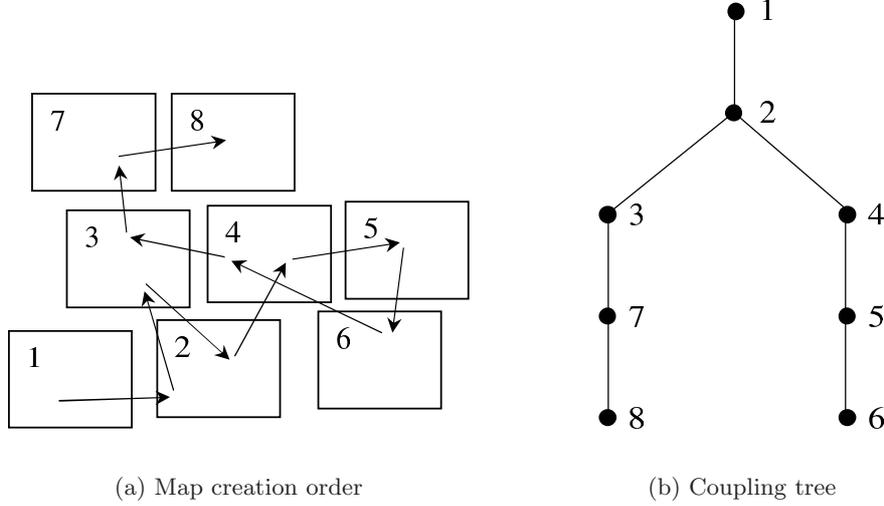


Figure 6.1: Submap coupling tree. New submaps are coupled to their parent submap according to the vehicle pose at creation. In this example, from [29, 30], the coupling tree was constructed from the following order-of-events. Create maps 1, 2, then 3, reenter map 2, create maps 4, 5, and 6, reenter map 4, then 3, create maps 7 and 8.

pose between submaps 4 and 6 is the vector sum of 4-to-5 and 5-to-6.

$${}^{S_4}\hat{\mathbf{x}}_{S_6} = \mathbf{f}({}^{S_4}\hat{\mathbf{x}}_{S_5}, {}^{S_5}\hat{\mathbf{x}}_{S_6}) = \begin{bmatrix} {}^{S_4}\hat{x}_{S_5} + {}^{S_5}\hat{x}_{S_6} \cos {}^{S_4}\hat{\phi}_{S_5} - {}^{S_5}\hat{y}_{S_6} \sin {}^{S_4}\hat{\phi}_{S_5} \\ {}^{S_4}\hat{y}_{S_5} + {}^{S_5}\hat{x}_{S_6} \sin {}^{S_4}\hat{\phi}_{S_5} + {}^{S_5}\hat{y}_{S_6} \cos {}^{S_4}\hat{\phi}_{S_5} \\ {}^{S_4}\hat{\phi}_{S_5} + {}^{S_5}\hat{\phi}_{S_6} \end{bmatrix}$$

$${}^{S_4}\mathbf{P}_{S_6} = \nabla \mathbf{f}_{S_4 \mathbf{x}_{S_5}} {}^{S_4}\mathbf{P}_{S_5} \nabla \mathbf{f}_{S_4 \mathbf{x}_{S_5}} + \nabla \mathbf{f}_{S_5 \mathbf{x}_{S_6}} {}^{S_5}\mathbf{P}_{S_6} \nabla \mathbf{f}_{S_5 \mathbf{x}_{S_6}}^T$$

where the Jacobians $\nabla \mathbf{f}_{S_4 \mathbf{x}_{S_5}}$ and $\nabla \mathbf{f}_{S_5 \mathbf{x}_{S_6}}$ are found as follows.

$$\nabla \mathbf{f}_{S_4 \mathbf{x}_{S_5}} = \left. \frac{\partial \mathbf{f}}{\partial {}^{S_4}\mathbf{x}_{S_5}} \right|_{({}^{S_4}\hat{\mathbf{x}}_{S_5}, {}^{S_5}\hat{\mathbf{x}}_{S_6})}$$

$$\nabla \mathbf{f}_{S_5 \mathbf{x}_{S_6}} = \left. \frac{\partial \mathbf{f}}{\partial {}^{S_5}\mathbf{x}_{S_6}} \right|_{({}^{S_4}\hat{\mathbf{x}}_{S_5}, {}^{S_5}\hat{\mathbf{x}}_{S_6})}$$

The relative pose between submaps 4 and 6 can be calculated using two separate Jacobians because the individual coupling estimates are uncorrelated. However, a problem arises when this approach is applied to relative pose calculations involving a “common root.” For example, the relative pose between submaps 3 and 4 is found from the coupling estimates 2-to-3 and 2-to-4. In [29, 30], this calculation is performed in a separable fashion similar to the example above (i.e., assuming ${}^{S_2}\mathbf{x}_{S_3}$ and ${}^{S_2}\mathbf{x}_{S_4}$ are independent) when, in fact, the

coupling estimates are correlated. This error³ is rectified in [140], as described below.

A further issue with this method is that it lacks convergence at a global level; the coupling estimates remain static throughout. This is particularly troublesome for cycle detection and loop closure since the uncertainties accumulated over a large cycle of submaps cannot be reduced even after the loop has been completed (e.g., consider a cycle between submaps 4 and 8, the coupling estimate would remain the path 4-2-3-7-8). In other words, the coupling estimate between the two submaps connected by the cycle will possess a large and irreducible uncertainty equal to the sum of monotonic linkages connecting them.

6.1.5 Constrained Relative Submap Filter

The *constrained relative submap filter* (CRSF) [140] is similar in structure to the submap hierarchy in [29, 30], but presents significant advantages in terms of correlation estimation and global convergence.

The CRSF is similar to the method in [29, 30] in the following ways. It consists of independent submaps performing traditional SLAM and coupled via “monotonic linkage.” When a new submap is created, its origin is defined by the current vehicle pose, and an estimate of this pose is held by its “parent” submap. Calculation of the relative pose between two non-adjacent submaps is performed via the coupling tree. Both methods are consistent at a global level, avoid accumulated non-linearities, and require $O(n^2)$ computation and $O(Nn^2)$ storage (where N is the number of submaps and n is the number of features per submap).

The CRSF differs from the method in [29, 30] in the way it stores the coupling estimates, and the consequences of this storage approach. Rather than extract the relative pose estimate from the parent submap state, the coupling estimate (and its correlations to the parent map) is retained as follows.

$${}_{S_1}\hat{\mathbf{x}}_a = [{}_{S_1}\hat{\mathbf{x}}_{S_2}^T, {}_{S_1}\hat{\mathbf{x}}_m^T]^T$$

$${}_{S_1}\mathbf{P}_a = \begin{bmatrix} {}_{S_1}\mathbf{P}_{S_2} & {}_{S_1}\mathbf{P}_{S_2m} \\ {}_{S_1}\mathbf{P}_{S_2m}^T & {}_{S_1}\mathbf{P}_m \end{bmatrix}$$

When the vehicle subsequently returns to the parent map, the vehicle estimate is reinitialised using geometric constraints, so that the state becomes ${}_{S_1}\mathbf{x}_a = [{}_{S_1}\mathbf{x}_v^T, {}_{S_1}\mathbf{x}_{S_2}^T, {}_{S_1}\mathbf{x}_m^T]^T$. There are two important consequences of this formulation. First, if the parent submap later creates a second child submap, the correlations between the two child submaps are known; this corrects the error in [29, 30] for “common root” calculations. Second, as the vehicle performs SLAM within the parent submap, the map converges and the coupling estimate also converges through its correlations to the map. Thus, the overall CRSF map experiences global convergence while maintaining statistical independence between submaps.

A further advantage of the CRSF discussed in [140] is that geometric constraints can eventually be applied between the submaps so that it merges into a consistent monolithic map. This operation is shown to be almost equivalent to optimal SLAM implementations, but with considerable improvements in computation and reduction of non-linearities.

³The common root “error” in [29, 30] does not actually constitute an inconsistency; rather it results in an over-conservative estimate of the coupling uncertainty (see Section 6.2.6).

The CRSF in its present form is limited in terms of convergence. At a local level, the couplings between submaps can converge to a lower limit defined by the correlations between the coupling estimate and the parent submap. More importantly, at the global level, convergence is restricted by monotonic linkage and, as in [29, 30], loop closure is unable to reduce the uncertainty between two submaps, which is specified by the coupling tree path.

The CRSF is a promising alternative to the NCFM approach presented in this chapter, particularly if the loop closure convergence problem can be solved. In general, this requires an investigation into whether a coupling estimate can be formed between two existing submaps, which shortcuts the coupling tree, without having to propagate geometric constraints throughout the entire map (i.e., without having to correlate all submaps).

6.2 Network Coupled Feature Maps

This section presents a new submap framework called *network coupled feature maps* (NCFM).⁴ The basic structure of NCFM is shown in Figure 6.2, where independent submaps are connected by a network (or graph) of coupling estimates—these being the relative pose between submap coordinate frames. This concept is similar to the submap methods in [29, 30] and [140], but with the important difference that the coupling estimates are not restricted to monotonic linkages. The implementation and consequences of this difference are discussed later in this section.

This section presents the following basic mechanisms for SLAM within the NCFM framework.

- Coupling estimate equations. These equations for manipulating coupling estimates are key to global map management as they enable pose estimation across submaps and along coupling-paths connecting non-adjacent submaps.
- Map traversal. This includes traversal within and across existing submaps, with particular emphasis on consistent vehicle reinitialisation during submap transition.
- Submap creation. At the boundaries of the existing map, new submaps are created and coupled to neighbouring submaps. This procedure is developed for the case where the vehicle pose becomes the new coordinate origin, and also for when the origin is offset from the vehicle.
- Coupling convergence. Transitions between submaps allow for iterative improvement of their coupling estimates. A method for consistent coupling estimation is presented that produces a globally convergent map while retaining independent submaps. This

⁴The NCFM framework was first introduced in [4], where it was used primarily to demonstrate the reliability of batch data association in situations where the vehicle pose was highly uncertain. In May 2002, a method very similar to the NCFM framework was introduced at an oral presentation at Workshop W4, IEEE International Conference on Robotics and Automation (ICRA 2002). This approach, the *Atlas framework*, develops many of the same concepts given in this chapter. However, the author was made aware of this seminar only after submission of this thesis in August, and written details of the approach [18] only became available in October 2002. Thus, the NCFM methodology presented here was not influenced by the Atlas framework in conception or particulars.

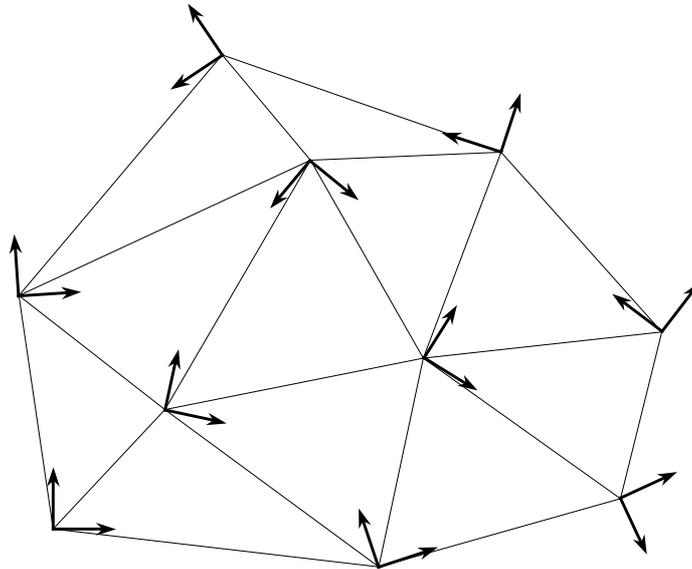


Figure 6.2: Network coupled feature map structure. The coordinate axes represent the origins of independent submaps. Neighbouring submaps are coupled by estimates of the relative pose between their coordinate origins.

method tends to correlate coupling estimates attached to any one submap, and the consequences of these correlations on global consistency are also discussed.

6.2.1 NCFM Overview

The NCFM structure consists of a local submap level and a global topological level. At the local level, each submap is a feature map defined with respect to a local coordinate frame. At the global level, each network link defines a coupling estimate of the relative pose between two submap coordinate origins.

Within a local submap, a robot performs traditional stochastic SLAM. The submap state vector is augmented with the vehicle state, and the normal SLAM operations are performed (e.g., prediction, observation, feature addition). When moving to another area, the vehicle state is initialised in the neighbouring submap using the coupling information connecting the two submaps, and so the robot transitions from one submap state vector to another. The vehicle state may even be maintained simultaneously in multiple neighbouring submaps. However, each submap is built independently from its neighbours and no information is ever transferred from one submap to another.

The coupling estimates between submaps are maintained so that they converge as their associated submaps converge. They enable consistent pose estimation across submaps, and effect map convergence at a global level.

The NCFM framework facilitates two basic navigation paradigms: topological-metric, and (conventional) global navigation. With topological-metric navigation, the vehicle might be located at a certain pose in submap A , and its goal location might be in submap D ,

which is connected to A via submaps B and C . The submap coupling estimates, therefore, describe a relative connecting path between the vehicle and its goal. Alternatively, for global navigation, a particular submap is nominated as the “base” frame, and poses relative to other submaps are converted to the base coordinate system via the topology of coupling estimates. Thus, NCFM can be used as though it were a traditional monolithic map.

Existing submap methods have solved the problem of efficiency for stochastic SLAM. However, the best of these solutions address either convergence (i.e., the RLR, with global base-feature coupling) or non-linearity (i.e., CRSF, with relative submap coupling), but not both. The NCFM framework presents a combined solution to efficiency, convergence and non-linearity.

In terms of efficiency, NCFM SLAM requires $O(Nn^2)$ storage and $O(n^2)$ computation, where N is the number of submaps and n is the number of features per submap. To convert the robot’s local pose to a global estimate takes $O(N)$ coupling evaluations between the current and base submaps. Note, this complexity assumes the coupling path is already known, and excludes the complexity of the connecting path algorithm (e.g., shortest path between two nodes [39]).

Non-linearity and bias are not significant issues for small-scale SLAM,⁵ but their cumulative effects are problematic for large-scale global maps, particularly in connection with loop closure. NCFM performs all map building at a local submap level so that error corrections are small and the effects of non-linearity and bias do not accumulate beyond submap bounds. The calculation of a “global” pose estimate, with respect to a base frame, is still subject to cumulative errors, but these are non-critical. First, the global estimate is never used in the actual SLAM update, and so does not effect the consistency of the map. And second, as the individual submaps converge, the local estimation problem becomes increasingly linear, and the global estimate accordingly becomes more linear.

Eventually biases may cause even small-scale SLAM to become inconsistent. The NCFM framework cannot prevent this but, by containing biases within submaps, serves to make the effects less pronounced and less detrimental.

Convergence at a submap level is the same as for traditional SLAM, and the coupling estimates converge as their associated submaps converge. Therefore, the map also converges at a global level. The lower limit of global convergence, as the features within each submap become fully correlated, is conjectured to approach that of traditional SLAM.

6.2.2 Coupling Estimate Equations

The coupling estimate of submap frame S_2 with respect to submap frame S_1 is denoted ${}^{S_1}\hat{\mathbf{x}}_{S_2}$ with covariance ${}^{S_1}\mathbf{P}_{S_2}$. In this case, S_1 is referred to as the *base* coordinate frame. Determining the pose of a particular submap (or of the vehicle) relative to a non-adjacent

⁵Non-linearity and bias problems fall into three basic categories. First, non-linear transformations cause Gaussian distributions to become non-Gaussian, and to be underestimated by a Gaussian approximation. These inconsistencies can be easily offset by adding stabilising noise to the process and observation models. Second, linearised error correction may be divergent if the error is large (i.e., if the first-order Taylor expansion is not a reasonable approximation to the non-linear function). This problem gives rise to the basic small-error assumption of the EKF. Third, persistent biases, which result from non-linear transformations and modelling errors, may cause the system to converge to a biased steady-state. This means that the system may eventually become inconsistent even with the addition of stabilising noise.

base frame is an important operation for a variety of navigation tasks, and is crucial for performing cycle detection (as described in Section 6.3). Three equations for coupling manipulation are presented here—coupling summation, coupling inversion and common root coupling—which permit recursive cross-submap calculations.

Coupling summation is the vector addition of two coupling estimates ${}^{S_1}\mathbf{x}_{S_2}$ and ${}^{S_2}\mathbf{x}_{S_3}$ to obtain the coupling ${}^{S_1}\mathbf{x}_{S_3}$. This operation can be used recursively to determine the pose of an entity with respect to a non-adjacent base frame.

$${}^{S_1}\hat{\mathbf{x}}_{S_3} = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_{S_2}, {}^{S_2}\hat{\mathbf{x}}_{S_3}) = \begin{bmatrix} {}^{S_1}\hat{x}_{S_2} + {}^{S_2}\hat{x}_{S_3} \cos {}^{S_1}\hat{\phi}_{S_2} - {}^{S_2}\hat{y}_{S_3} \sin {}^{S_1}\hat{\phi}_{S_2} \\ {}^{S_1}\hat{y}_{S_2} + {}^{S_2}\hat{x}_{S_3} \sin {}^{S_1}\hat{\phi}_{S_2} + {}^{S_2}\hat{y}_{S_3} \cos {}^{S_1}\hat{\phi}_{S_2} \\ {}^{S_1}\hat{\phi}_{S_2} + {}^{S_2}\hat{\phi}_{S_3} \end{bmatrix} \quad (6.1)$$

$${}^{S_1}\mathbf{P}_{S_3} = \nabla\mathbf{f}_{S_1\mathbf{x}_{S_2}} {}^{S_1}\mathbf{P}_{S_2} \nabla\mathbf{f}_{S_1\mathbf{x}_{S_2}}^T + \nabla\mathbf{f}_{S_2\mathbf{x}_{S_3}} {}^{S_2}\mathbf{P}_{S_3} \nabla\mathbf{f}_{S_2\mathbf{x}_{S_3}}^T \quad (6.2)$$

where the Jacobians $\nabla\mathbf{f}_{S_1\mathbf{x}_{S_2}}$ and $\nabla\mathbf{f}_{S_2\mathbf{x}_{S_3}}$ are given by

$$\nabla\mathbf{f}_{S_1\mathbf{x}_{S_2}} = \left. \frac{\partial\mathbf{f}}{\partial S_1\mathbf{x}_{S_2}} \right|_{S_1\hat{\mathbf{x}}_{S_2}} = \begin{bmatrix} 1 & 0 & -{}^{S_2}\hat{x}_{S_3} \sin {}^{S_1}\hat{\phi}_{S_2} - {}^{S_2}\hat{y}_{S_3} \cos {}^{S_1}\hat{\phi}_{S_2} \\ 0 & 1 & {}^{S_2}\hat{x}_{S_3} \cos {}^{S_1}\hat{\phi}_{S_2} - {}^{S_2}\hat{y}_{S_3} \sin {}^{S_1}\hat{\phi}_{S_2} \\ 0 & 0 & 1 \end{bmatrix} \quad (6.3)$$

$$\nabla\mathbf{f}_{S_2\mathbf{x}_{S_3}} = \left. \frac{\partial\mathbf{f}}{\partial S_2\mathbf{x}_{S_3}} \right|_{S_2\hat{\mathbf{x}}_{S_3}} = \begin{bmatrix} \cos {}^{S_1}\hat{\phi}_{S_2} & -\sin {}^{S_1}\hat{\phi}_{S_2} & 0 \\ \sin {}^{S_1}\hat{\phi}_{S_2} & \cos {}^{S_1}\hat{\phi}_{S_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

Given a connecting path between the base frame and the frame representing the object of interest, a “global” estimate of the object can be found. For example, in Figure 6.3 the base frame S_1 and the vehicle are connected by the path 1-2-3-v of piece-wise couplings ${}^{S_1}\mathbf{x}_{S_2}$, ${}^{S_2}\mathbf{x}_{S_3}$, and ${}^{S_3}\mathbf{x}_v$. A global vehicle estimate is obtained by applying Equation 6.1 twice to determine first ${}^{S_1}\mathbf{x}_{S_3}$ then ${}^{S_1}\mathbf{x}_v$.

Note that the covariance summation in Equation 6.2 is calculated with two separate Jacobians, which implies that the coupling estimates ${}^{S_1}\hat{\mathbf{x}}_{S_2}$ and ${}^{S_1}\hat{\mathbf{x}}_{S_3}$ are independent (i.e., uncorrelated). This is true given the coupling estimates when the submaps are first created (see Section 6.2.4), but is generally not the case following the convergence operations shown in Section 6.2.5. The issue of treating correlated coupling estimates as though they were uncorrelated is addressed in Section 6.2.6 (with the verdict being that the result is consistent and, in fact, conservative).

Coupling inversion is the calculation of the pose of frame S_1 with respect to frame S_2 , when given the pose of frame S_2 with respect to frame S_1 . That is, it allows the switching of base coordinate frames.

$${}^{S_2}\hat{\mathbf{x}}_{S_1} = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_{S_2}) = \begin{bmatrix} -{}^{S_1}\hat{x}_{S_2} \cos {}^{S_1}\hat{\phi}_{S_2} - {}^{S_1}\hat{y}_{S_2} \sin {}^{S_1}\hat{\phi}_{S_2} \\ {}^{S_1}\hat{x}_{S_2} \sin {}^{S_1}\hat{\phi}_{S_2} - {}^{S_1}\hat{y}_{S_2} \cos {}^{S_1}\hat{\phi}_{S_2} \\ -{}^{S_1}\hat{\phi}_{S_2} \end{bmatrix} \quad (6.5)$$

$${}^{S_2}\mathbf{P}_{S_1} = \nabla\mathbf{f}_{S_1\mathbf{x}_{S_2}} {}^{S_1}\mathbf{P}_{S_2} \nabla\mathbf{f}_{S_1\mathbf{x}_{S_2}}^T \quad (6.6)$$

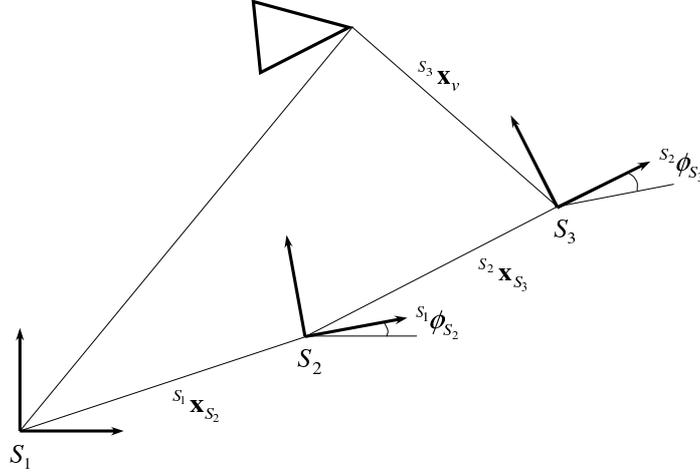


Figure 6.3: Coupling summation. The vehicle pose with respect to the base frame S_1 is found by recursive application of Equation 6.1. First ${}^{S_1}\mathbf{x}_{S_3} = \mathbf{f}({}^{S_1}\mathbf{x}_{S_2}, {}^{S_2}\mathbf{x}_{S_3})$ then ${}^{S_1}\mathbf{x}_v = \mathbf{f}({}^{S_1}\mathbf{x}_{S_3}, {}^{S_3}\mathbf{x}_v)$.

where the Jacobian $\nabla \mathbf{f}_{S_1 \mathbf{x}_{S_2}}$ is given by

$$\nabla \mathbf{f}_{S_1 \mathbf{x}_{S_2}} = \left. \frac{\partial \mathbf{f}}{\partial {}^{S_1}\mathbf{x}_{S_2}} \right|_{S_1 \hat{\mathbf{x}}_{S_2}} = \begin{bmatrix} -\cos {}^{S_1}\hat{\phi}_{S_2} & -\sin {}^{S_1}\hat{\phi}_{S_2} & {}^{S_1}\hat{x}_{S_2} \sin {}^{S_1}\hat{\phi}_{S_2} - {}^{S_1}\hat{y}_{S_2} \cos {}^{S_1}\hat{\phi}_{S_2} \\ \sin {}^{S_1}\hat{\phi}_{S_2} & -\cos {}^{S_1}\hat{\phi}_{S_2} & {}^{S_1}\hat{x}_{S_2} \cos {}^{S_1}\hat{\phi}_{S_2} + {}^{S_1}\hat{y}_{S_2} \sin {}^{S_1}\hat{\phi}_{S_2} \\ 0 & 0 & -1 \end{bmatrix} \quad (6.7)$$

Common root coupling is the calculation of ${}^{S_2}\hat{\mathbf{x}}_{S_3}$ given ${}^{S_1}\hat{\mathbf{x}}_{S_2}$ and ${}^{S_1}\hat{\mathbf{x}}_{S_3}$. That is, given the pose of two frames with respect to a common base frame, obtain the relative pose between the two frames.

$${}^{S_2}\hat{\mathbf{x}}_{S_3} = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_{S_2}, {}^{S_1}\hat{\mathbf{x}}_{S_3}) = \begin{bmatrix} ({}^{S_1}\hat{x}_{S_3} - {}^{S_1}\hat{x}_{S_2}) \cos {}^{S_1}\hat{\phi}_{S_2} + ({}^{S_1}\hat{y}_{S_3} - {}^{S_1}\hat{y}_{S_2}) \sin {}^{S_1}\hat{\phi}_{S_2} \\ -({}^{S_1}\hat{x}_{S_3} - {}^{S_1}\hat{x}_{S_2}) \sin {}^{S_1}\hat{\phi}_{S_2} + ({}^{S_1}\hat{y}_{S_3} - {}^{S_1}\hat{y}_{S_2}) \cos {}^{S_1}\hat{\phi}_{S_2} \\ {}^{S_1}\hat{\phi}_{S_3} - {}^{S_1}\hat{\phi}_{S_2} \end{bmatrix} \quad (6.8)$$

$${}^{S_2}\mathbf{P}_{S_3} = \nabla \mathbf{f}_{S_1 \mathbf{x}_{S_2}} {}^{S_1}\mathbf{P}_{S_2} \nabla \mathbf{f}_{S_1 \mathbf{x}_{S_3}}^T + \nabla \mathbf{f}_{S_1 \mathbf{x}_{S_3}} {}^{S_1}\mathbf{P}_{S_3} \nabla \mathbf{f}_{S_1 \mathbf{x}_{S_2}}^T \quad (6.9)$$

where the Jacobians $\nabla \mathbf{f}_{S_1 \mathbf{x}_{S_2}}$ and $\nabla \mathbf{f}_{S_1 \mathbf{x}_{S_3}}$ are given by

$$\nabla \mathbf{f}_{S_1 \mathbf{x}_{S_2}} = \left. \frac{\partial \mathbf{f}}{\partial {}^{S_1}\mathbf{x}_{S_2}} \right|_{S_1 \hat{\mathbf{x}}_{S_2}} = \begin{bmatrix} -\cos {}^{S_1}\hat{\phi}_{S_2} & -\sin {}^{S_1}\hat{\phi}_{S_2} & -({}^{S_1}\hat{x}_{S_3} - {}^{S_1}\hat{x}_{S_2}) \sin {}^{S_1}\hat{\phi}_{S_2} + ({}^{S_1}\hat{y}_{S_3} - {}^{S_1}\hat{y}_{S_2}) \cos {}^{S_1}\hat{\phi}_{S_2} \\ \sin {}^{S_1}\hat{\phi}_{S_2} & -\cos {}^{S_1}\hat{\phi}_{S_2} & -({}^{S_1}\hat{x}_{S_3} - {}^{S_1}\hat{x}_{S_2}) \cos {}^{S_1}\hat{\phi}_{S_2} - ({}^{S_1}\hat{y}_{S_3} - {}^{S_1}\hat{y}_{S_2}) \sin {}^{S_1}\hat{\phi}_{S_2} \\ 0 & 0 & -1 \end{bmatrix} \quad (6.10)$$

$$\nabla \mathbf{f}_{S_1 \mathbf{x}_{S_3}} = \left. \frac{\partial \mathbf{f}}{\partial {}^{S_1}\mathbf{x}_{S_3}} \right|_{S_1 \hat{\mathbf{x}}_{S_3}} = \begin{bmatrix} \cos {}^{S_1}\hat{\phi}_{S_2} & \sin {}^{S_1}\hat{\phi}_{S_2} & 0 \\ -\sin {}^{S_1}\hat{\phi}_{S_2} & \cos {}^{S_1}\hat{\phi}_{S_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.11)$$

The three coupling manipulation equations can be summarised using vector notation as follows.

- Coupling summation.

$${}^{S_1}\mathbf{x}_{S_3} = {}^{S_1}\mathbf{x}_{S_2} + \mathbf{R} {}^{S_2}\mathbf{x}_{S_3} \quad (6.12)$$

- Coupling inversion.

$${}^{S_2}\mathbf{x}_{S_1} = \mathbf{R}^{-1}(-{}^{S_1}\mathbf{x}_{S_2}) \quad (6.13)$$

- Common root coupling.

$${}^{S_2}\mathbf{x}_{S_3} = \mathbf{R}^{-1}({}^{S_1}\mathbf{x}_{S_3} - {}^{S_1}\mathbf{x}_{S_2}) \quad (6.14)$$

where \mathbf{R} is the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos {}^{S_1}\hat{\phi}_{S_2} & -\sin {}^{S_1}\hat{\phi}_{S_2} & 0 \\ \sin {}^{S_1}\hat{\phi}_{S_2} & \cos {}^{S_1}\hat{\phi}_{S_2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

6.2.3 Map Traversal

Traversing a local submap is performed using traditional stochastic SLAM. The size of these submaps is presumed reasonably small in comparison to the vehicle's sensing range so that minor inconsistencies in the map update (due to model non-linearities and biases) do not accumulate significantly, and can be absorbed by stabilising noise.

When traversing the edge regions of a submap S_1 , attempts are made to associate observed data with the adjacent submap S_2 . For batch association purposes, the predicted pose of the vehicle is projected into S_2 using the common root coupling equation ${}^{S_2}\hat{\mathbf{x}}_v = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_{S_2}, {}^{S_1}\hat{\mathbf{x}}_v)$ as shown in Equation 6.8. This is concatenated with the submap feature estimates as follows.

$${}^{S_2}\hat{\mathbf{x}}_a = \begin{bmatrix} {}^{S_2}\hat{\mathbf{x}}_v \\ {}^{S_2}\hat{\mathbf{x}}_m \end{bmatrix} \quad (6.15)$$

$${}^{S_2}\mathbf{P}_a = \begin{bmatrix} {}^{S_2}\mathbf{P}_v & \mathbf{0} \\ \mathbf{0} & {}^{S_2}\mathbf{P}_m \end{bmatrix} \quad (6.16)$$

Batch association is performed using the tracking version of CCDA given in Section 3.3.4. If an acceptable association set is found, then a new vehicle track is initialised in the second submap. It is essential that no information is transferred between the two submaps as this would introduce correlations between them; rather the vehicle pose is initialised with zero *a priori* knowledge beyond the given data association information.

The method proposed in this thesis for initialisation uses the *insert and observe* procedure (see Appendix C), and further details regarding vehicle initialisation are provided in Appendix C.2. A nominal value ${}^{S_2}\hat{\mathbf{x}}_v$ for the vehicle location is the same as in Equation 6.15.

The prior state for S_2 is given by the concatenation of the vehicle pose guess, with infinite uncertainty,⁶ and the submap feature estimates.

$$S_2 \hat{\mathbf{x}}_a^- = \begin{bmatrix} S_2 \hat{\mathbf{x}}_v \\ S_2 \hat{\mathbf{x}}_m \end{bmatrix} \quad (6.17)$$

$$S_2 \mathbf{P}_a^- = \begin{bmatrix} \alpha \mathbf{I}_v & \mathbf{0} \\ \mathbf{0} & S_2 \mathbf{P}_m \end{bmatrix} \quad (6.18)$$

A posterior estimate of the vehicle pose in the second submap is then obtained by applying the normal observation update equations, as given in Section 2.2.3, for each of the associated observations from the batch association process.

A word of caution at this point. If both submaps continue to maintain a vehicle track, it is imperative that no observation information is used in both submaps. That is, if both submaps contain common features (i.e., features representing the same physical objects), observations of these features may be used in one or other of the submaps, but not both. While this information reuse would not be a concern if the submaps were permanently independent, it is a problem when they are both used to perform coupling estimation in Section 6.2.5. (Note, since batch data association is performed separately in each submap using the same observation data set, common features are found when a particular measurement is assigned to a landmark in both submaps.)

An alternative mechanism for vehicle initialisation is presented in [140], where observations are compiled in a small local map and subsequently applied to the submap using geometric constraints.

6.2.4 Submap Creation

As the vehicle traverses the boundaries of the existing map, new submaps are created to represent new regions of the environment. Conventionally, the vehicle pose has been used to depict the origin of a new submap at its creation. Thus, the new submap S_2 is created as

$$S_2 \hat{\mathbf{x}}_a = \begin{bmatrix} S_2 \mathbf{0}_v \\ S_2 \hat{\mathbf{x}}_m \end{bmatrix} \quad (6.19)$$

$$S_2 \mathbf{P}_a = \begin{bmatrix} S_2 \mathbf{0}_v & \mathbf{0} \\ \mathbf{0} & S_2 \mathbf{P}_m \end{bmatrix} \quad (6.20)$$

and the coupling to its parent submap S_1 is given by the estimated vehicle pose $S_1 \hat{\mathbf{x}}_v$ at this moment. Note, this estimate may be extracted from S_1 (as in [29, 30]), or may be retained in the parent state estimate (as in the CRSF).

In certain applications, it may be desirable to create a new submap with origin offset from the current vehicle pose. For example, in Figure 6.4 the new submap is offset so that its axis is aligned with the parent frame and the submap region can be specified by geometric bounds (i.e., square submaps for efficient region coverage with minimal overlap). The two operations for offset submap creation are (i) calculation of the coupling estimate $S_1 \hat{\mathbf{x}}_{S_2}$ (with

⁶The insert and observe method uses a suitably large number α as an approximation for infinity (see Appendix C).

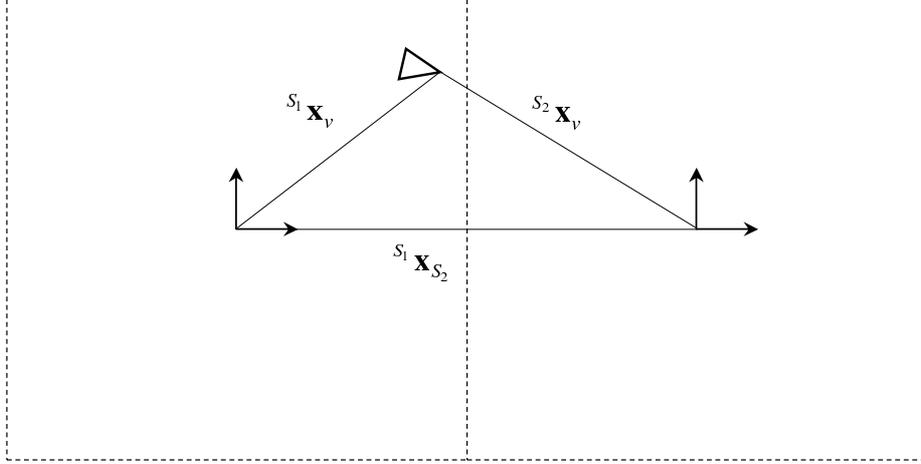


Figure 6.4: Submap creation with offset origin. A rigid body transform is performed with respect to the vehicle frame to produce a coordinate origin with a nominal coupling ${}^{S_1}\mathbf{x}_{S_2}$ to the parent frame.

uncertainty ${}^{S_1}\mathbf{P}_{S_2}$), and (ii) calculation of the augmented estimate ${}^{S_2}\hat{\mathbf{x}}_a = [{}^{S_2}\hat{\mathbf{x}}_v^T, {}^{S_2}\hat{\mathbf{x}}_m^T]^T$ with respect to the offset coordinate frame.

Given the current vehicle pose estimate ${}^{S_1}\hat{\mathbf{x}}_v$ and a nominal new submap origin ${}^{S_1}\mathbf{x}_{S_2}$ with respect to the parent submap, the offset submap frame relative to the vehicle is found from the common root coupling equation ${}^v\mathbf{x}_{S_2} = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_v, {}^{S_1}\mathbf{x}_{S_2})$ as in Equation 6.8. Note, ${}^v\mathbf{x}_{S_2}$ is a rigid-body extension with respect to the vehicle such that ${}^v\mathbf{P}_{S_2} = \mathbf{0}$. The coupling estimate is subsequently calculated using a coupling summation equation ${}^{S_1}\hat{\mathbf{x}}_{S_2} = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_v, {}^v\mathbf{x}_{S_2})$ as in Equation 6.1, such that the estimated pose of the new submap is the original nominal value and the coupling uncertainty is found according to Equation 6.2 as follows.

$$\begin{aligned} {}^{S_1}\mathbf{P}_{S_2} &= \nabla\mathbf{f}_{S_1\mathbf{x}_v} {}^{S_1}\mathbf{P}_v \nabla\mathbf{f}_{S_1\mathbf{x}_v}^T + \nabla\mathbf{f}_{v\mathbf{x}_{S_2}} {}^v\mathbf{0}_{S_2} \nabla\mathbf{f}_{v\mathbf{x}_{S_2}}^T \\ &= \nabla\mathbf{f}_{S_1\mathbf{x}_v} {}^{S_1}\mathbf{P}_v \nabla\mathbf{f}_{S_1\mathbf{x}_v}^T \end{aligned}$$

The vehicle pose estimate relative to the new submap frame is found as a coupling inversion ${}^{S_2}\hat{\mathbf{x}}_v = \mathbf{f}({}^v\hat{\mathbf{x}}_{S_2})$ as in Equation 6.5, and has zero uncertainty ${}^{S_2}\mathbf{P}_v = \mathbf{0}$. At creation, the map features are assumed to be uncorrelated to the vehicle and from each other; thus, feature estimates ${}^v\hat{\mathbf{x}}_i$ can be independently transformed from the vehicle frame to the new submap frame using a common root calculation ${}^{S_2}\hat{\mathbf{x}}_i = \mathbf{f}({}^v\hat{\mathbf{x}}_{S_2}, {}^v\hat{\mathbf{x}}_i)$. The resulting augmented state estimate ${}^{S_2}\hat{\mathbf{x}}_a$ is therefore block-diagonal.

Finally, the new submap is connected to other submaps that are adjacent to the parent submap and possess region boundaries adjacent to the new submap. These couplings are determined using a common root equation ${}^{S_2}\hat{\mathbf{x}}_{S_3} = \mathbf{f}({}^{S_1}\hat{\mathbf{x}}_{S_2}, {}^{S_1}\hat{\mathbf{x}}_{S_3})$.

6.2.5 Coupling Convergence

In the course of traversing submap boundary regions, vehicle tracks may be maintained independently in multiple adjacent submaps (with care that observation information is used

at most once). The information from these multiple tracks can be used to improve the coupling estimates between submaps without correlating the submaps themselves. Convergence of the coupling estimates facilitates convergence of the map at a global level.

The method for performing coupling convergence is two stage. First, a new estimate of the coupling between two submaps is obtained and, second, this estimate is fused with the prior coupling estimate.

A new coupling estimate between two submaps is found using the insert-and-observe procedure as follows. First, the two augmented state vectors $^{S_1}\hat{\mathbf{x}}_a$ and $^{S_2}\hat{\mathbf{x}}_a$ are concatenated along with an initial “guess” of the coupling estimate $^{S_1}\hat{\mathbf{x}}_{S_2}$. The coupling guess is assigned infinite uncertainty as follows.

$$\hat{\mathbf{x}}_c^- = [^{S_1}\hat{\mathbf{x}}_{S_2}^T, ^{S_1}\hat{\mathbf{x}}_v^T, ^{S_1}\hat{\mathbf{x}}_m^T, ^{S_2}\hat{\mathbf{x}}_v^T, ^{S_2}\hat{\mathbf{x}}_m^T]^T \quad (6.21)$$

$$\mathbf{P}_c^- = \begin{bmatrix} \alpha\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & ^{S_1}\mathbf{P}_v & ^{S_1}\mathbf{P}_{vm} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & ^{S_1}\mathbf{P}_{vm}^T & ^{S_1}\mathbf{P}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ^{S_2}\mathbf{P}_v & ^{S_2}\mathbf{P}_{vm} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & ^{S_2}\mathbf{P}_{vm}^T & ^{S_2}\mathbf{P}_m \end{bmatrix} \quad (6.22)$$

Second, a constrained estimate of the concatenated state is generated by applying two types of perfect “virtual” observations. The first of these is that the vehicle in both submaps is the same, $\mathbf{z}_v = \mathbf{0}$ where

$$\hat{\mathbf{z}}_v = \mathbf{h}_v(\hat{\mathbf{x}}_c) = ^{S_1}\hat{\mathbf{x}}_v - (^{S_1}\hat{\mathbf{x}}_{S_2} + \mathbf{R}^{S_2}\hat{\mathbf{x}}_v) \quad (6.23)$$

and \mathbf{R} is the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos ^{S_1}\hat{\phi}_{S_2} & -\sin ^{S_1}\hat{\phi}_{S_2} & 0 \\ \sin ^{S_1}\hat{\phi}_{S_2} & \cos ^{S_1}\hat{\phi}_{S_2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This constraint has the following innovation and innovation covariance.

$$\nu = \mathbf{z}_v - \hat{\mathbf{z}}_v = -\mathbf{h}_v(\hat{\mathbf{x}}_c^-) \quad (6.24)$$

$$\mathbf{S} = \nabla\mathbf{h}_{\mathbf{x}_c}\mathbf{P}_c^-\nabla\mathbf{h}_{\mathbf{x}_c}^T \quad (6.25)$$

where the Jacobian $\nabla\mathbf{h}_{\mathbf{x}_c} = \left. \frac{\partial\mathbf{h}_v}{\partial\mathbf{x}_c} \right|_{\hat{\mathbf{x}}_c}$.

The second constraint type arises from the existence of common features between the two submaps. These are found from batch data association when, for a given set of observations, the same measurement is assigned to a feature in both maps; (this measurement is then used to update only one of these features). Thus, for common features $^{S_1}\mathbf{x}_i$ and $^{S_2}\mathbf{x}_j$, the constraint is $\mathbf{z}_{ij} = \mathbf{0}$ where

$$\hat{\mathbf{z}}_{ij} = \mathbf{h}_{ij}(\hat{\mathbf{x}}_c) = ^{S_1}\hat{\mathbf{x}}_i - (^{S_1}\hat{\mathbf{x}}_{S_2} + \mathbf{R}^{S_2}\hat{\mathbf{x}}_j) \quad (6.26)$$

Here, $^{S_1}\mathbf{x}_{S_2}$ refers to just the (x, y) part of the coupling estimate, and \mathbf{R} is the matrix

$$\mathbf{R} = \begin{bmatrix} \cos ^{S_1}\hat{\phi}_{S_2} & -\sin ^{S_1}\hat{\phi}_{S_2} \\ \sin ^{S_1}\hat{\phi}_{S_2} & \cos ^{S_1}\hat{\phi}_{S_2} \end{bmatrix}$$

The innovation $\nu = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}$ and innovation covariance can then be derived, and these “virtual” observations can be used to produce a posterior estimate of the concatenated state vector via the normal EKF update equations. From this result, the new coupling estimate ${}^{S_1}\hat{\mathbf{x}}_{S_2}$, ${}^{S_1}\mathbf{P}_{S_2}$ is extracted.

There are three points to note from the above process. First, the value of the initial “guess” for ${}^{S_1}\hat{\mathbf{x}}_{S_2}$ is simply the prior coupling estimate. Second, the concatenated vector is formed with *copies* of the two submaps; these coupling calculations are separate and isolated from the independent submap SLAM operations. Third, for efficiency, only those features common to both submaps need to be copied into the concatenated state (non-common feature exert no influence on the result).

Furthermore, if the two submaps are parent-child and the original coupling estimate is retained in the parent state estimate (as for the CRSF), then the following concatenated prior can be used instead of Equation 6.22.

$$\mathbf{P}_c^- = \begin{bmatrix} {}^{S_1}\mathbf{P}_{S_2} & {}^{S_1}\mathbf{P}_{S_2v} & {}^{S_1}\mathbf{P}_{S_2m} & \mathbf{0} & \mathbf{0} \\ {}^{S_1}\mathbf{P}_{S_2v}^T & {}^{S_1}\mathbf{P}_v & {}^{S_1}\mathbf{P}_{vm} & \mathbf{0} & \mathbf{0} \\ {}^{S_1}\mathbf{P}_{S_2m}^T & {}^{S_1}\mathbf{P}_{vm}^T & {}^{S_1}\mathbf{P}_m & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & {}^{S_2}\mathbf{P}_v & {}^{S_2}\mathbf{P}_{vm} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & {}^{S_2}\mathbf{P}_{vm}^T & {}^{S_2}\mathbf{P}_m \end{bmatrix} \quad (6.27)$$

This prior incorporates the initial knowledge of the coupling estimate as retained in the parent submap. However, it is important to note that the coupling estimate used in Equation 6.27 is part of the submap state and separate from the coupling estimate obtained from the concatenation process. In other words, there are two versions of coupling estimate: (i) the estimate contained in the parent submap from the child’s creation, and (ii) the coupling estimates obtained from constraining the concatenated state. It is also worth noticing that the advantage of the prior in Equation 6.27 over the insert-and-observe approach in Equation 6.22 may prove to be marginal as the submaps themselves converge, and probably does not warrant the added complexity.

Having obtained a new coupling estimate ${}^{S_1}\hat{\mathbf{x}}_{S_2}^{new}$, the new estimate is fused with the prior coupling estimate ${}^{S_1}\hat{\mathbf{x}}_{S_2}^{old}$ using the *covariance intersection* [77]. This permits monotonically convergent estimation when an unknown degree of correlation exists between the prior and new estimates.⁷ For notational convenience, let $\mathbf{x}_a = {}^{S_1}\mathbf{x}_{S_2}^{old}$ and $\mathbf{x}_b = {}^{S_1}\mathbf{x}_{S_2}^{new}$; the posterior estimate is found from the covariance intersection equations as follows.

$$\mathbf{P}_c = (w\mathbf{P}_a^{-1} + (1-w)\mathbf{P}_b^{-1})^{-1} \quad (6.28)$$

$$\hat{\mathbf{x}}_c = \mathbf{P}_c (w\mathbf{P}_a^{-1}\hat{\mathbf{x}}_a + (1-w)\mathbf{P}_b^{-1}\hat{\mathbf{x}}_b) \quad (6.29)$$

where the scalar variable $0 \leq w \leq 1$ is chosen to minimise some criterion (e.g., determinant, trace) of the posterior covariance matrix \mathbf{P}_c .

A consequence of the coupling convergence method shown here is that the resulting coupling estimate is correlated to both submaps and, moreover, to every other coupling

⁷The covariance intersection is hardly necessary if the coupling estimate involves common feature constraints. This tends to produce “comparable” uncertainty estimates, where ${}^{S_1}\mathbf{P}_{S_2}^{old} - {}^{S_1}\mathbf{P}_{S_2}^{new}$ is positive semi-definite, and the new estimate simply replaces the old. The covariance intersection is most useful when only vehicle pose constraints are involved.

connected to either submap. This is a serious issue for the coupling equations given in Section 6.2.2, which assume uncorrelated coupling estimates. The question of whether these equations still produce consistent results if the coupling estimates are correlated is addressed below.

6.2.6 Consequences of Non-independent Couplings

This section investigates the properties of pair-wise correlated⁸ relative pose estimates, and is applicable to the coupling equations in Section 6.2.2 and also to the sensor-based dead reckoning algorithm in Example C.1. The outcome of this investigation suggests that treating the coupling estimates as independent, when they are actually pair-wise correlated, produces consistent and conservative results. This is an important finding as it means coupling equations can be used directly on converging coupling estimates, and indicates that NCFM SLAM may attain near-optimal convergence limits at a global level.

The general case for correlated information is as follows. Given two independent random vectors \mathbf{a} and \mathbf{b} , and a model relating them to a third vector $\mathbf{c} = \mathbf{f}(\mathbf{a}, \mathbf{b})$, the covariance of the estimate $\hat{\mathbf{c}}$ can be calculated as follows.

$$\mathbf{P}_c = \nabla \mathbf{f}_a \mathbf{P}_a \nabla \mathbf{f}_a^T + \nabla \mathbf{f}_b \mathbf{P}_b \nabla \mathbf{f}_b^T \quad (6.30)$$

where the Jacobians are $\nabla \mathbf{f}_a = \frac{\partial \mathbf{f}}{\partial \mathbf{a}}|_{(\hat{\mathbf{a}}, \hat{\mathbf{b}})}$ and $\nabla \mathbf{f}_b = \frac{\partial \mathbf{f}}{\partial \mathbf{b}}|_{(\hat{\mathbf{a}}, \hat{\mathbf{b}})}$. If \mathbf{a} and \mathbf{b} are correlated, however, Equation 6.30 is invalid and may produce an inconsistent result, although a consistent estimate can be found using covariance intersection methods.

$$\mathbf{P}_c = \nabla \mathbf{f}_a \frac{\mathbf{P}_a}{w} \nabla \mathbf{f}_a^T + \nabla \mathbf{f}_b \frac{\mathbf{P}_b}{1-w} \nabla \mathbf{f}_b^T \quad (6.31)$$

where w is a scalar between 0 and 1. In particular, if \mathbf{P}_a and \mathbf{P}_b are each inflated by a factor of two, their substitution in Equation 6.30 produces a consistent estimate.

A basic extension of this concept is to say that, for a path of pair-wise correlated couplings, a consistent ‘‘coupling summation’’ estimate can be obtained by first inflating each coupling covariance by two. Clearly, this expansion results in a greatly inflated uncertainty in the final estimate.

At this point, it is important to remark that ignoring correlations does not necessarily imply inconsistency; the action of ignoring correlations can lead to either inconsistent (optimistic) estimates or suboptimal (conservative) estimates. Consider the following 1-D example $c = a + b$, where the variables a and b possess variances A and B , respectively, and unknown correlation D . That is, the vector $[a, b]^T$ has covariance

$$\begin{bmatrix} A & D \\ D & B \end{bmatrix}$$

The *true* variance of c , therefore, is $C = A + B + 2D$ while, ignoring correlations, the estimated variance is $C = A + B$. The difference between the estimated and true variance is

⁸For a coupling path connecting two coordinate frames, pair-wise correlation means that each coupling estimate is correlated to its preceding and succeeding coupling estimates only. This is true for any non-closed path in the NCFM framework.

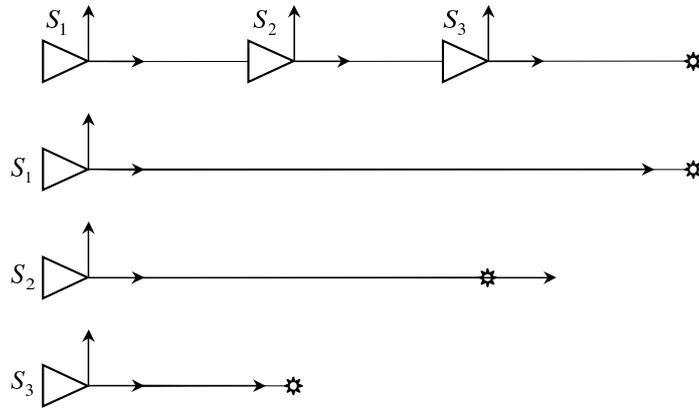


Figure 6.5: Coupling correlations. Three submap origins in a global coordinate frame are shown in the top diagram; the location of a single landmark is shown also. The three lower diagrams depict estimates of the landmark relative to each frame (arrow) and its true location (asterisk). The coupling estimates $S_1 \hat{x}_{S_2}$ and $S_2 \hat{x}_{S_3}$ are correlated by the error in S_2 .

$-2D$, which is positive-semi-definite if $D \leq 0$. This means that ignoring correlations will lead to inconsistent results if $D > 0$ but will give conservative results if $D < 0$.⁹ In the former case, consistent results can be obtained by inflating A and B by two, but in the latter case, the original variance values suffice.

It is now shown, by way of a 1-D example, that the correlations occurring in the coupling equations are always of the conservative variety. Thus, inflation of the coupling estimate uncertainties is unnecessary.

Example 6.1

One-dimensional coupling summation. This example considers a simple case where three submaps share the one feature. This might also be interpreted as sensor-based dead reckoning over three timesteps while observing a single landmark. The result shown here is equally valid for scenarios with multiple landmarks.

A single landmark L is represented in three submaps as shown in Figure 6.5. The estimated distances in each submap are $S_1 \hat{x}_L = d_1$, $S_2 \hat{x}_L = d_2$ and $S_3 \hat{x}_L = d_3$. The true relative distances are $d_1 + v_1$, $d_2 + v_2$ and $d_3 + v_3$, respectively, where v_1 , v_2 and v_3 are independent random variables. The coupling estimates $S_1 \hat{x}_{S_2}$ and $S_2 \hat{x}_{S_3}$ are given by

$$\begin{aligned} S_1 \hat{x}_{S_2} &= S_1 \hat{x}_L - S_2 \hat{x}_L = d_1 - d_2 \\ S_2 \hat{x}_{S_3} &= S_1 \hat{x}_L - S_2 \hat{x}_L = d_2 - d_3 \end{aligned}$$

while the true couplings are $d_1 + v_1 - d_2 - v_2$ and $d_2 + v_2 - d_3 - v_3$, respectively. Note, the two estimates are correlated by the error v_2 in S_2 . Coupling summation results in the following estimated coupling.

$$S_1 \hat{x}_{S_3} = S_1 \hat{x}_{S_2} + S_2 \hat{x}_{S_3} = d_1 - d_2 + d_2 - d_3 = d_1 - d_3$$

⁹In more intuitive terms, if $D > 0$ then a and b will tend to be either above or below their mean values together—exacerbating the error in c —while, if $D < 0$ then a and b will tend to occupy opposite sides of their means (e.g., a above and b below), and be self-correcting in c .

The true coupling is

$$d_1 + v_1 - d_2 - v_2 + d_2 + v_2 - d_3 - v_3 = d_1 + v_1 - d_3 - v_3$$

Notice that the correlated error v_2 cancels out and the summation is self-correcting. If the couplings were uncorrelated, the summation would have combined the uncertainty of four random variables— v_1, v_2, v_2', v_3 —but, since they are correlated, it involves only two.

Another way to explain this example is to say that the vector $[S_1 \hat{x}_L, S_2 \hat{x}_L, S_3 \hat{x}_L]^T$ has covariance

$$\mathbf{P}_L = \begin{bmatrix} S_1 \mathbf{P}_L & 0 & 0 \\ 0 & S_2 \mathbf{P}_L & 0 \\ 0 & 0 & S_3 \mathbf{P}_L \end{bmatrix}$$

The coupling estimates are calculated from this as follows.

$$\begin{bmatrix} S_1 \hat{x}_{S_2} \\ S_2 \hat{x}_{S_3} \end{bmatrix} = \mathbf{F} \begin{bmatrix} S_1 \hat{x}_L \\ S_2 \hat{x}_L \\ S_3 \hat{x}_L \end{bmatrix}$$

where \mathbf{F} is the transition matrix

$$\mathbf{F} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

The covariance of the coupling estimates, therefore, is obtained as

$$\mathbf{P}_S = \mathbf{F} \mathbf{P}_L \mathbf{F}^T = \begin{bmatrix} S_1 \mathbf{P}_L + S_2 \mathbf{P}_L & -S_2 \mathbf{P}_L \\ -S_2 \mathbf{P}_L & S_2 \mathbf{P}_L + S_3 \mathbf{P}_L \end{bmatrix}$$

Notice that the off-diagonals of \mathbf{P}_S are negative, which means that the coupling summation $S_1 \hat{x}_{S_3} = S_1 \hat{x}_{S_2} + S_2 \hat{x}_{S_3}$ is conservative if the couplings are treated as uncorrelated. That is, the uncorrelated estimate covariance $S_1 \mathbf{P}_L + 2S_2 \mathbf{P}_L + S_3 \mathbf{P}_L$ is always greater than the true correlated covariance $S_1 \mathbf{P}_L + S_3 \mathbf{P}_L$.

This example demonstrates that, for 1-D coupling summation, assuming uncorrelated coupling estimates (i.e., ignoring correlations) will produce a conservative summed estimate.

From the above 1-D example, it is conjectured that coupling summation, assuming uncorrelated coupling estimates, is conservative also for higher dimensions. This is suggested empirically by the dead reckoning results in Section 3.4, where the noise between sequential scans is quite high but the global summed pose is extremely accurate. It appears that the scan matching errors tend to cancel for each pair of relative pose estimates.

An intuitive argument for the error cancelling properties of correlated coupling estimates is given by considering the limiting case. In the limit, each submap becomes (internally) fully correlated, so that its features approach a lower bound with respect to the coordinate frame, and become perfectly known relative to each other. This may be visualised as a rigid lattice of features that is still flexible relative to its coordinate origin. A coupling estimate, derived from common feature constraints between two submaps, defines a transform so that

the two submaps are perfectly aligned, even though their relative origins are uncertain. Similarly, a path of coupling estimates is perfectly aligned; they produce a sequence of rigid lattices connected together by common feature constraints. The locations of the intervening submap origins are uncertain, but this is irrelevant since the coupling estimates are defined by the relative feature locations, which are perfectly known.

If this reasoning is correct, then it would appear that the lower bound feature uncertainty in each submap cancels out for all intermediate submaps in a coupling path, and the global accuracy of NCFM approaches almost the same lower limit as full SLAM. In other words, the uncertainty of the intermediate couplings is dependent solely on the *rigidity* of the connected submaps, and not the uncertainty of their coordinate origins. Only the coordinate frame uncertainties of the beginning and end submaps remain relevant.

Therefore, it is suggested that the true accuracy of NCFM converges to near that of traditional SLAM, but the NCFM covariance estimate will be over-conservative since it incorporates the uncertainty of each submap origin.

6.3 Cycle Detection within the NCFM Framework

For all SLAM methodologies, the most difficult challenge is cycle detection. This section presents a complete strategy for robust cycle detection and confirmation within the NCFM framework, so that loop closure becomes a feasible proposition in extensive environments with massive accumulated vehicle pose uncertainty.

The process of cycle detection is implemented as a three stage operation.

- First pass detection. A efficient search is performed to find all submaps not adjacent to the current submap that may be within the vehicle field-of-view (i.e., predicted submap visibility).
- Second pass detection. Batch data association determines the existence of possible cycles to any of the candidate submaps.
- Confirmation. Each cycle hypothesis is tracked until satisfying reasonable criteria as a confirmed cycle. The coupling linkages for the cycle are then added to the map.

This strategy essentially combines the cycle detection mechanisms of topological maps (i.e., place recognition and rehearsal) with the added restriction of metric pose constraints. Nevertheless, these checks are insufficient to guarantee against false cycles, and this section concludes with a discussion of conditions in which cycle detection may fail.

6.3.1 First Pass Detection

First pass search is performed to determine the possible visibility of every submap not adjacent to the current submap. This search checks if any part of a non-adjacent submap region intersects with the vehicle field-of-view region, and efficiently culls the number of submaps valid for the second pass detection phase.

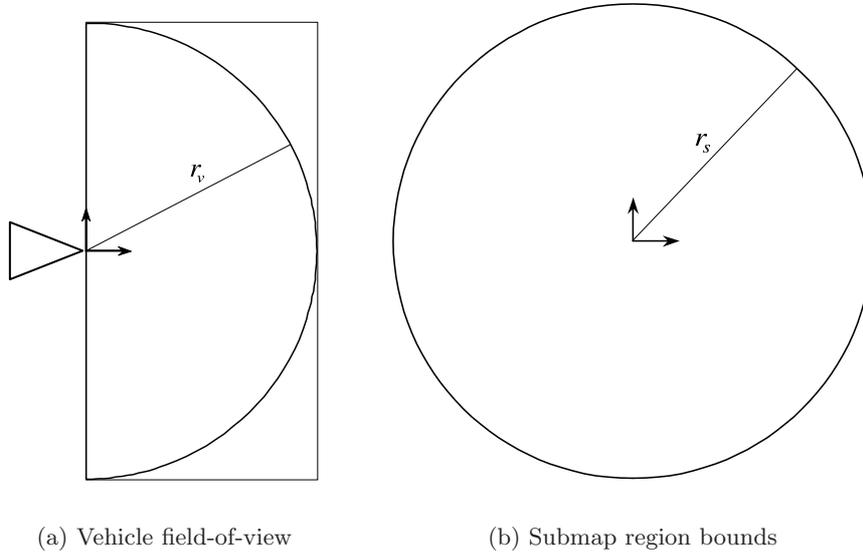


Figure 6.6: Vehicle and submap visibility regions. The vehicle field-of-view is semi-circular with radius r_v ; this region is bounded by a rectangular box. The submap region is bounded by a circle of radius r_s .

For each non-adjacent submap S_i , the relative pose with respect to the vehicle ${}^v\mathbf{x}_{S_i}$ is found via coupling summation along the shortest connecting path.¹⁰ The submaps are checked in breadth-first order radiating from the current submap. This serves the dual purpose of (i) finding the (approximately) shortest connecting path to each submap, and (ii) permitting storage of the cumulative coupling sum estimates at each level of the search tree (i.e., avoiding recalculation for subsequent levels).

The possible visibility test described below assumes that the vehicle field-of-view is bounded by a rectangular box $\{(0, r_v), (r_v, r_v), (r_v, -r_v), (0, -r_v)\}$ as shown in Figure 6.6(a), and the submap region is bounded about its origin by a circle of radius r_s , as shown in Figure 6.6(b).

The first step in calculating visibility is to define the $n\sigma$ (n -sigma) uncertainty ellipse of the submap origin with respect to the vehicle. (The value of n sets a threshold on the uncertainty search-space; typically n equals two or three.) These equations do not involve the relative orientation ${}^v\phi_{S_i}$, so let $\hat{\mathbf{x}} = [{}^v\hat{x}_{S_i}, {}^v\hat{y}_{S_i}]^T$ with covariance \mathbf{P} . Let the elements of \mathbf{P} be denoted as

$$\mathbf{P} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} a & d \\ d & b \end{bmatrix} \quad (6.32)$$

The eigenvalues of \mathbf{P} are found from the characteristic equation $|\mathbf{P} - \lambda\mathbf{I}| = 0$. That is,

¹⁰The path chosen between a submap and the vehicle is not critical, and the most accurate solution would be to calculate ${}^v\hat{\mathbf{x}}_{S_i}$, ${}^v\mathbf{P}_{S_i}$ along all combinations of connecting paths and fuse them using the covariance intersect. A practical solution is to simply choose the shortest connecting path, which usually corresponds to the minimum uncertainty path.

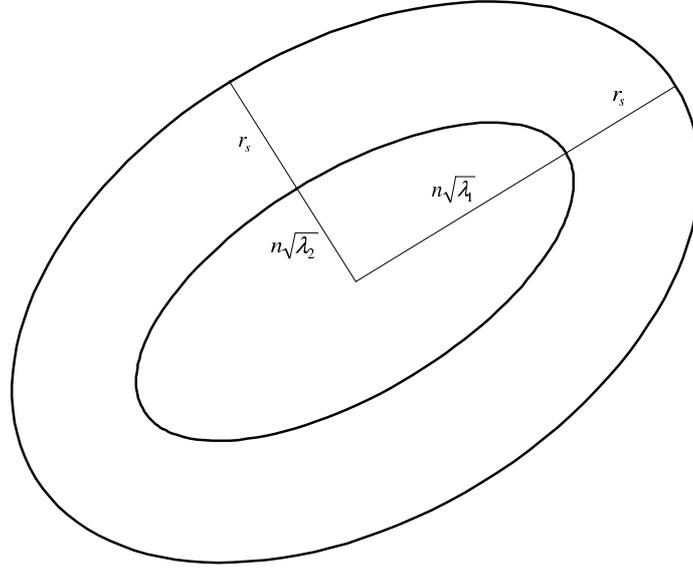


Figure 6.7: Submap visibility region. The inner ellipse represents the $n\sigma$ uncertainty ellipse of the submap origin with respect to the vehicle. This region is extended by the submap radius r_s to define the region over which the submap might be visible.

$(a - \lambda)(b - \lambda) - d^2 = 0$, which has two roots

$$\lambda = \frac{a + b \pm \sqrt{(a - b)^2 + 4d^2}}{2} \quad (6.33)$$

The eigenvectors corresponding to the principal axes of \mathbf{P} are then found as

$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \begin{bmatrix} \lambda + d - b \\ \lambda + d - a \end{bmatrix} \quad (6.34)$$

These eigenvectors are normalised by their 2-norm $\|\mathbf{e}\| = \sqrt{e_x^2 + e_y^2}$ and scaled by $n\sqrt{\lambda}$ to produce the principal axes of an $n\sigma$ ellipse. This ellipse is depicted as the inner ellipse of Figure 6.7, which is subsequently expanded by the submap radius r_s to define the extent of the region over which the submap is possibly visible with respect to the vehicle. The principal eigenvectors of this outer ellipse are calculated from the unscaled-eigenvectors in Equation 6.34 as follows.

$$\mathbf{e}_s = (n\sqrt{\lambda} + r_s) \frac{\mathbf{e}}{\|\mathbf{e}\|} \quad (6.35)$$

The two scaled eigenvector \mathbf{e}_{s1} and \mathbf{e}_{s2} will subsequently be referred to as \mathbf{e}_1 and \mathbf{e}_2 .

Having defined the elliptical region of possible submap visibility, the test for possible visibility becomes an intersection test between this ellipse and the vehicle field-of-view rectangle. The intersection test requires some mathematical preliminaries relating the ellipse eigenvectors to Cartesian coordinates.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e_{1x} \cos \theta + e_{2x} \sin \theta \\ e_{1y} \cos \theta + e_{2y} \sin \theta \end{bmatrix} \quad (6.36)$$

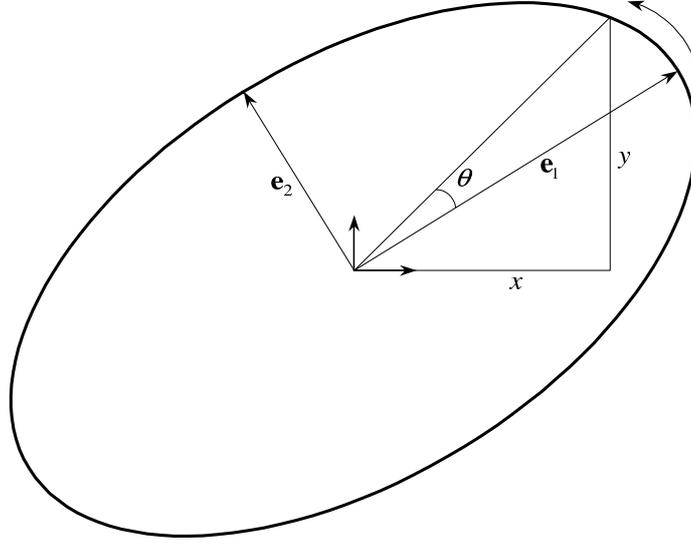


Figure 6.8: Cartesian ellipse coordinates. The (x, y) coordinates of the ellipse are obtained as a function of the principal eigenvectors \mathbf{e}_1 , \mathbf{e}_2 and the angle θ .

where θ is the angle from the positive axis of \mathbf{e}_1 in the direction of the positive axis of \mathbf{e}_2 as shown in Figure 6.8. The two solutions for θ from Equation 6.36 are

$$\theta = \arctan \frac{xe_{2_x}^2 \pm e_{1_x}e_{2_x}\sqrt{k_x}}{xe_{1_x}e_{2_x} \mp e_{2_x}^2\sqrt{k_x}} \quad (6.37)$$

$$\theta = \arctan \frac{ye_{2_y}^2 \pm e_{1_y}e_{2_y}\sqrt{k_y}}{ye_{1_y}e_{2_y} \mp e_{2_y}^2\sqrt{k_y}} \quad (6.38)$$

where $k_x = e_{1_x}^2 + e_{2_x}^2 - x^2$ and $k_y = e_{1_y}^2 + e_{2_y}^2 - y^2$. Note, θ is not defined (has imaginary roots) for $k_x < 0$ or $k_y < 0$; thus, the Cartesian extremities of the ellipse are $x = \pm\sqrt{e_{1_x}^2 + e_{2_x}^2}$ and $y = \pm\sqrt{e_{1_y}^2 + e_{2_y}^2}$.

Note, the location of the ellipse centre relative to the vehicle is $\hat{\mathbf{x}}$, but for these calculations it is simpler to make the ellipse centre the origin and shift the vehicle by $-\hat{\mathbf{x}}$. Thus, the ellipse is defined by the eigenvectors \mathbf{e}_1 and \mathbf{e}_2 , and the rectangle is given by coordinates $\{(-\hat{x}, r_v - \hat{y}), (r_v - \hat{x}, r_v - \hat{y}), (r_v - \hat{x}, -r_v - \hat{y}), (-\hat{x}, -r_v - \hat{y})\}$.

The test for ellipse-rectangle intersection is performed in three stages and, if any stage succeeds, the subsequent stages are not needed. The first is to find whether any line of the box intersects with the ellipse; the second tests if the box is enclosed by the ellipse; and the third tests if the ellipse is enclosed by the box.¹¹

1. Box-ellipse intersection test. This test is performed on each line segment of the rectangle. Here, the vertical segment $\{(x_p, y_p), (x_p, y_q)\}$, $y_p < y_q$ is shown as an example.

¹¹The third test, ellipse-in-box, is typically not required as the size of the submap region is usually greater than the vehicle field-of-view.

- (a) Check whether the vertical line $x = x_p$ intersects the ellipse by satisfying the criterion $x_p^2 \leq e_{1_x}^2 + e_{2_x}^2$ (n.b., use the form $y_p^2 \leq e_{1_y}^2 + e_{2_y}^2$ for horizontal lines). If not, the test fails; repeat test with a different line segment.
 - (b) Calculate the two solutions θ_1 and θ_2 for Equation 6.37 where $x = x_p$ (n.b., use Equation 6.38 for horizontal lines).
 - (c) Calculate y_1 and y_2 for θ_1 and θ_2 , respectively, using Equation 6.36. These are the two ellipse intersect points for the line $x = x_p$.
 - (d) If $y_p \leq y_1 \leq y_q$ or $y_p \leq y_2 \leq y_q$, then this test returns true.
2. Box-in-ellipse test. This test involves choosing any corner of the rectangle (x_p, y_p) .
 - (a) Check the vertical line $x = x_p$ intersects the ellipse by satisfying the criterion $x_p^2 \leq e_{1_x}^2 + e_{2_x}^2$. If not, the test fails.
 - (b) Perform steps (b) and (c) of the box-ellipse intersection test.
 - (c) If $y_1 \leq y_p \leq y_2$ then this test returns true.
 3. Ellipse-in-box test. This is a trivial point-in-box test which returns true if the ellipse centre $(0, 0)$ lies within the rectangle bounds.

The non-adjacent submaps that meet the ellipse-rectangle intersection criteria are marked as potentially visible and are then subjected to the second pass detection process.

6.3.2 Second Pass Detection

The second pass stage of cycle detection performs batch data association within each candidate submap. If sufficient associations are obtained between the current observation set and a submap S_i , then a *cycle hypothesis* is formed. This process is analogous to topological place recognition.

The expected vehicle pose ${}^{S_i}\hat{\mathbf{x}}_v$ with respect to the submap S_i is found via coupling summation. Thus, the augmented submap state for batch association incorporates pose constraints (though these are not correlated to the map states) as follows.

$${}^{S_i}\hat{\mathbf{x}}_a = \begin{bmatrix} {}^{S_i}\hat{\mathbf{x}}_v \\ {}^{S_i}\hat{\mathbf{x}}_m \end{bmatrix} \quad (6.39)$$

$${}^{S_i}\mathbf{P}_a = \begin{bmatrix} {}^{S_i}\mathbf{P}_v & \mathbf{0} \\ \mathbf{0} & {}^{S_i}\mathbf{P}_m \end{bmatrix} \quad (6.40)$$

The submap state is associated to the vehicle observation set using the tracking CCDA algorithm presented in Section 3.3.4. Note, the feature grouping procedure, as discussed in Section 5.3, is not required for submap cycle detection because correlated feature grouping is inherent within the NCFM submap structure.

Each successful batch association, including multiple association hypotheses within a single submap, forms a cycle hypothesis. Each cycle hypothesis is tracked over a specified period and distance to ascertain its validity (i.e., to ensure it is not an artifact of environmental symmetry). This phase is termed cycle confirmation.

6.3.3 Cycle Confirmation

The cycle confirmation stage tracks a set of cycle hypotheses until a single hypothesis remains and continues for a reasonable time and distance. This process is analogous to topological rehearsal.

The objective of cycle confirmation is to protect against short-term symmetries in the environment, where different areas of the environment possess similar structural appearance over a limited region. A cycle hypothesis is rejected if tracking is lost—if batch association fails to produce sufficient correspondences as the vehicle moves through the region. When several cycle possibilities exist, tracking continues until at most one persists. The remaining hypothesis is accepted if it persists longer than a specified time period and vehicle traversal distance.

During the confirmation process, SLAM is continued in the current submap and its adjacencies, including the creation of new submaps. That is, map building and traversal carry on as though cycle detection were not in progress. However, in the “cycle hypothesis” submaps, the vehicle pose is tracked separately from the submap state. That is, the submap state is static and the vehicle performs *localisation* only as it confirms the cycle validity. When a cycle is eventually confirmed, the current submap is coupled to the cycle submap (and some of their neighbouring submaps are also connected). The coupling estimate is calculated using the coupling convergence mechanism described in Section 6.2.5. It may be beneficial even to merge the current submap with the cycle submap, using geometric constraints, if they possess sufficient overlap.

6.3.4 Pathological Symmetries

The cycle detection and confirmation strategy presented above is robust but cannot absolutely guarantee the rejection of false cycles. It is possible that symmetries in the environment may persist over a large enough area to admit an incorrect validation. This problem is most likely to occur in highly structured environments.

In general, symmetry-based cycle hypotheses will be rejected because they coexist with the correct cycle hypothesis, which will eventually become the only surviving hypothesis. However, there are two possible scenarios where false cycle rejection may fail. First, it may fail in dynamic environments if distinctive features are obscured by transient objects and the true hypothesis is rejected—leaving symmetry-based hypotheses unchallenged. Second, it may fail if the vehicle is exploring a new region of the environment that possesses symmetry with an old mapped region, and is close enough to this region to satisfy pose constraints. This second scenario has no “true” cycle and will be accepted if the symmetry is sufficiently persistent.

Consider, for example, the situation in Figure 6.9, where a vehicle traverses a large loop in a highly structured corridor environment. At the final position shown, the vehicle is about to enter a new corridor identical to the first. Because of its large pose uncertainty, the vehicle forms a cycle hypothesis with the original location (X) and, since the new corridor is not yet part of the map, the false cycle persists unchallenged until the end of the corridor. If the cycle confirmation distance is less than the corridor length, the false cycle will be added to the map.

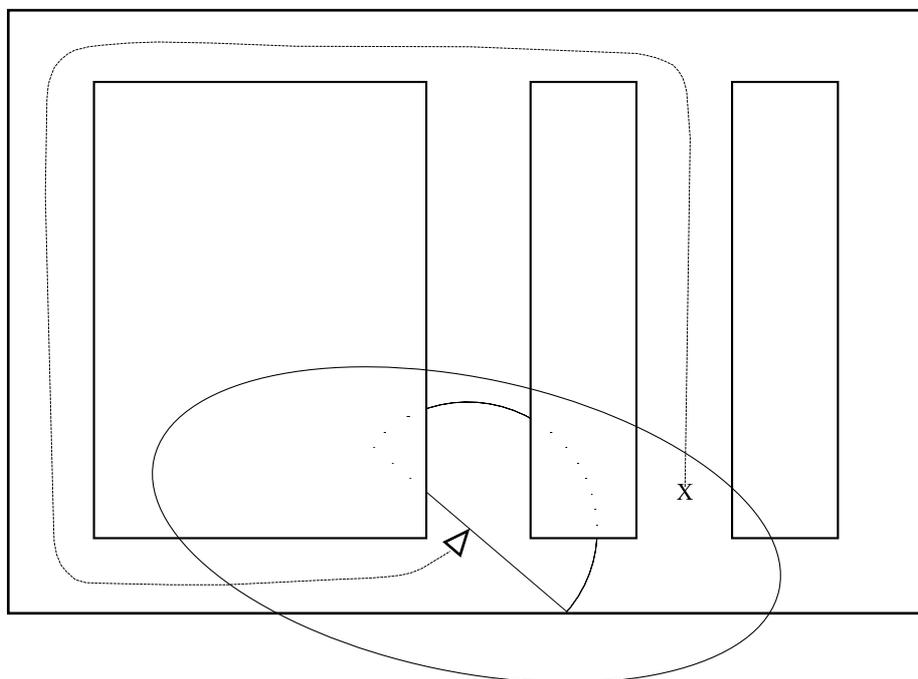


Figure 6.9: False cycle detection. This picture signifies a corridor environment where the robot starts at location X and travels along the trajectory shown. At the final pose, the true pose (triangle) and vehicle field of view (semicircle) are depicted, along with the pose uncertainty ellipse. The new corridor is structurally identical to the original (adjacent) corridor.

6.4 Application: Large-scale Outdoor NCFM SLAM

This application demonstrates the basic attributes of NCFM with an implementation of SLAM in the internal road environment. As a first-cut attempt at NCFM SLAM, this experiment incorporates two major simplifications to the method described in this chapter. First, each submap is represented only by the scan of features obtained at its creation, and remains static thereafter. Second, the coupling estimates between submaps are formed when a new submap is created or upon a cycle confirmation, and these also remain static afterwards. Note, as a result of these two restrictions, some rather *ad hoc* operations are included in the experimental details below.

The procedures demonstrated in this application are: map traversal and submap transition, submap creation, and cycle detection and confirmation. Aspects described in this chapter that are not demonstrated here are: SLAM within local submaps, coupling convergence, and multiple cycle hypotheses per submap.

6.4.1 Implementation Details

This experiment is feature-based, with the details for feature extraction being the same as given in Section 3.4.1. The vehicle prediction model is the same as in Section 3.4.2 and, if

tracking is temporarily lost with respect to the local submaps, the vehicle pose estimate is propagated using sensor-based dead reckoning.

Map traversal is simply a matter of tracking the features of whichever surrounding submaps are predicted visible, and relying on dead reckoning if none provide a sufficient number of associations. (Tracking may be lost intermittently because the single-scan submap representations do not tend to cover the environment very evenly.)

A new submap is created once the vehicle moves more than 15 metres from the origin of the nearest existing submap. The new submap is defined by the features extracted from the current laser scan, with the vehicle pose as its origin. Coupling estimates are then created between all submaps with origins closer than 40 metres.

A simple version of cycle detection is performed with each new laser scan. The first-pass detection does not consider the size of the submap or vehicle view regions, but simply looks for non-adjacent submaps with origins closer to the vehicle than a fixed NIS threshold. That is, $\hat{\mathbf{x}}\mathbf{P}\hat{\mathbf{x}} < \gamma_2$ where $\hat{\mathbf{x}}$, \mathbf{P} represent the (x, y) part of the relative pose estimate ${}^v\hat{\mathbf{x}}_{S_i}$, ${}^v\mathbf{P}_{S_i}$ for submap S_i . The second-pass detection stage performs tracking CCDA in each candidate submap and, for submaps yielding sufficient associations (≥ 7), cycle hypotheses are formed on the basis of maximum association set (i.e., maximum cliques). Thus, at most one cycle hypothesis is formed per candidate submap each timestep. These hypotheses are added to a hypothesis list provided each particular hypothesis does not already exist on the list.

Cycle hypotheses are tracked for a rehearsal period across three submaps. If tracking is lost, due to insufficient associations, for more than 10 scans in a row, the hypothesis is deleted. If a hypothesis survives rehearsal, and all other hypotheses are significantly younger, then the cycle is confirmed and the remaining hypotheses are deleted. Connections are made between the original cycle submap and the submap that was current when the cycle was *first* hypothesised; the later “current” submaps created during the rehearsal period are removed from the map.

6.4.2 Results

NCFM SLAM is performed over the same trajectory as presented in Section 5.4. The resultant map is displayed using two different global representations in Figures 6.10 and 6.11. These figures attempt to exhibit the coupling properties of the map, which permit very accurate pose estimates between closely connected submaps.

In Figure 6.10, the submap locations are shown at their estimated global positions when first created. It shows the accumulated uncertainty of latter submaps with respect to the first, even if they are quite close physically. It also depicts the major differences between predicted and estimated pose following cycle detection and confirmation, which are a primary source of non-linearity problems for traditional SLAM implementations.

The plot in Figure 6.11 shows exactly the same map, but with the submap locations estimated from breadth-first coupling summation. Summation along a breadth-first path means that a shortest path corresponding to an (approximately) minimum uncertainty estimate ${}^{S_1}\hat{\mathbf{x}}_{S_i}$ is obtained. The resulting global map gives a much better indication of its true structure, and demonstrates the improved certainty of short coupling paths compared to long circuitous paths.

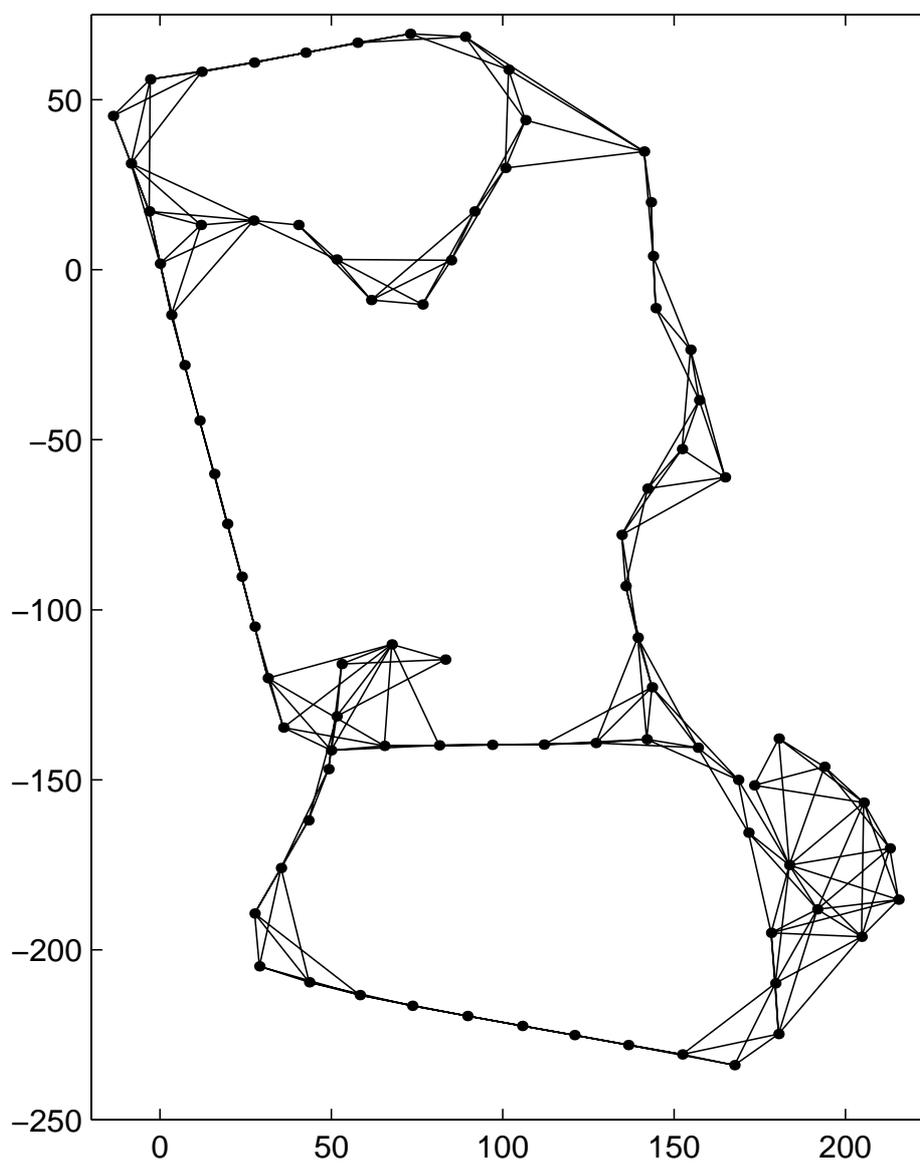


Figure 6.10: NCFM with submap origins plotted at their original global location estimates. This plot shows the large discontinuities between the predicted global pose and the estimated pose after cycle confirmation.

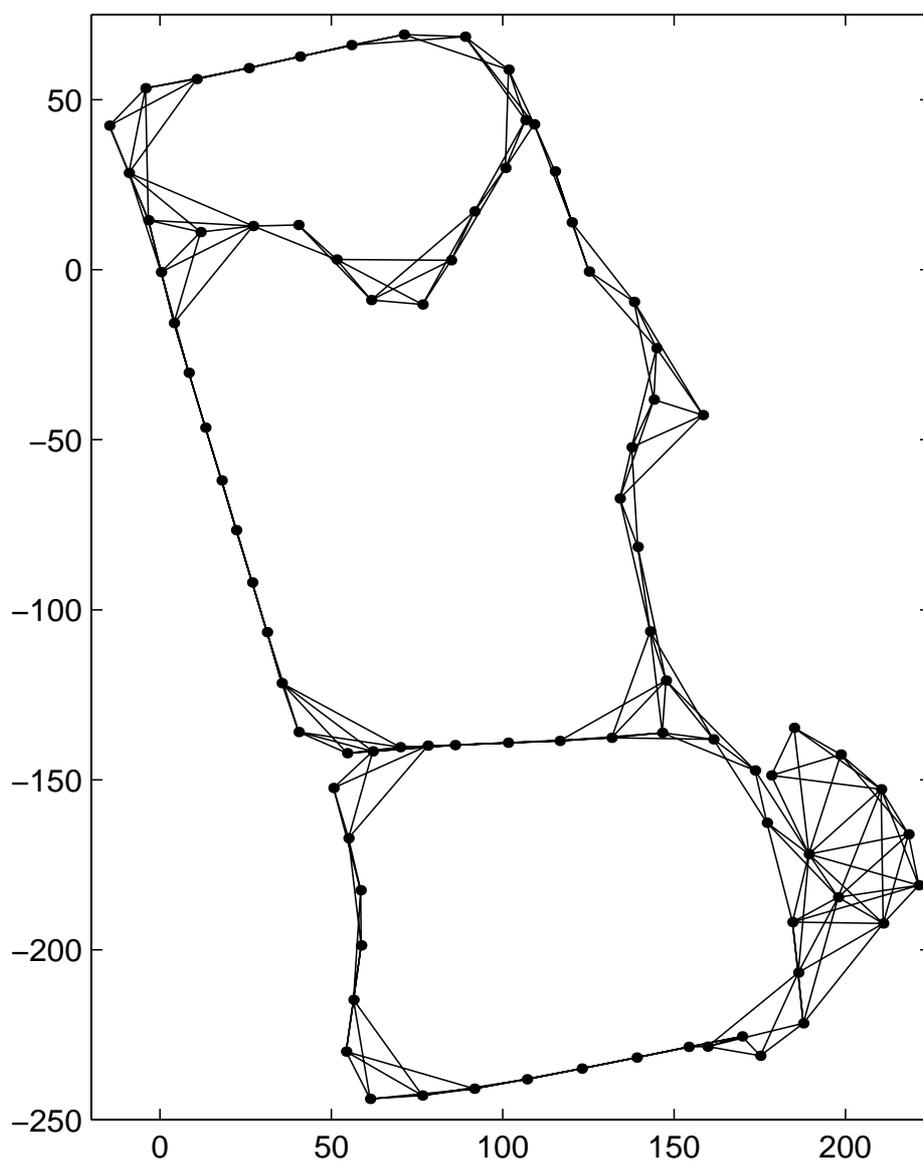


Figure 6.11: NCFM with submap origins plotted in breadth-first order. This plot results in an (approximately) minimum uncertainty global map with respect to the first submap.

6.5 Summary

This chapter presents the NCFM framework for SLAM in large-scale environments.

Current variants of submap-based stochastic SLAM are reviewed, with particular attention to their consistency during submap transitions. The DSM method, while shown empirically to give satisfactory results, lacks a theoretical proof of consistency. Also, DSM, the “two-level landmark” hierarchy, and the RLR, having global base coordinate systems, fail to address the non-linearity problems incurred by large loop closures. The “hierarchical local maps” and CRSF methods are both consistent and have similar structure, but CRSF provides significant advantages in the way it defines its submap couplings. Both methods are limited by monotonic linkage.

The NCFM framework is presented for SLAM operations of local traversal, submap transition, and submap creation. The map converges at a local (submap) level via traditional SLAM and at a global level by monotonic improvement of the coupling estimates. The consequent correlation between coupling estimates is believed to have an error-canceling effect, so that NCFM SLAM may even converge to a near-optimal lower limit.

A strategy for robust cycle detection is presented for NCFM. This involves three stages: first-pass search, second-pass detection, and cycle confirmation. The result is a method for loop closure that is resistant to all but the most pathological environmental symmetries.

A preliminary demonstration of the NCFM method is presented in a large-scale outdoor environment with challenging loop closures. Basic implementation is shown for submap traversal and creation and cycle detection, but stochastic SLAM at a local level and coupling estimate level remains to be empirically verified.

Chapter 7

Conclusion

This thesis attempts to address the theoretical and practical issues which previously limited SLAM implementations to small-to-medium scale environments with simple distinct landmarks. The primary difficulties concerning SLAM in complex large-scale environments are seen to be

- Reliable data association given large uncertainties in the vehicle position.
- The representation of landmarks that are not suited to simple geometric classification.
- Map management in terms of landmark addition and removal to avoid the long-term accumulation of clutter in the map.
- Map management in terms of submaps to permit computationally tractable (scalable) and mathematically consistent SLAM.
- The reliable detection of cycles (loops) in the map and consistent map update on loop closure.

This thesis presents solutions to these problems and verifies their practical utility through experimental applications in outdoor environments (i.e., sensor-based dead reckoning, *a priori* map localisation, traditional SLAM, submap SLAM).

This chapter summaries the contributions of this thesis and proposes a set of future directions for completing and extending this work.

7.1 Summary of Contributions

Four primary theoretical contributions are presented in this thesis: the CCDA algorithm, Gaussian sum scan correlation, mechanisms for feature management, and the NCFM framework. These contributions, and their practical implications, are reviewed below.

7.1.1 Combined Constraint Data Association

The CCDA algorithm considers a set of possible observation-map correspondences as a batch rather than individually. This method incorporates all available correlation information to

constrain association, and specifies sets of associations that are mutually compatible. A key property of CCDA is its ability to produce likely association sets even if the vehicle pose is entirely unknown.

An application of batch data association is to permit sensor-based dead reckoning, which estimates the ego-motion of the vehicle without odometry. This is of practical importance in rugged outdoor environments, where significant wheel-slip can compromise the assumptions of a kinematic vehicle model.

Contributions

- CCDA provides a solution to the problem of fragile data association, dramatically reducing reliance on multiple hypotheses tracking for reliability.
- CCDA offers the same constraint properties as the JCBB batch association method, but additionally permits data association with unknown vehicle pose. This is useful for initialising a vehicle within an existing map.

7.1.2 Sum of Gaussians Scan Correlation

Scan correlation—the alignment of unprocessed point data sets—is a valuable alternative to feature-based data association in environments not suited to geometric feature models. In this thesis, a Gaussian sum representation of point data is shown to facilitate Bayesian scan correlation. That is, it represents the data so that the ensuing relative pose estimate is obtained from a justifiable Bayesian likelihood function, derived from a stochastic model of the sensor uncertainty. The relative pose likelihood function is shown to be given by cross-correlation of the respective Gaussian sums.

The scan correlation method is demonstrated by applications of maximum-likelihood sensor-based dead reckoning and particle filter localisation.

Contributions

- Presentation of a representation and means to perform Bayesian scan correlation between unprocessed point data sets without feature extraction. This approach provides an accurate and consistent estimate of the relative pose uncertainty and does not involve data association.
- Development of a practical implementation of the scan correlation algorithm using a Gaussian sum representation, which permits accurate description of the sensor uncertainty and efficient calculation of the relative pose likelihood function.

7.1.3 Feature Management

The first contribution towards feature management is the compilation of useful existing techniques for feature initialisation and removal, notably the method of constrained initialisation in [140]. Two motivations are given for feature removal: control of map density and deletion of obsolete (non-existent) features.

Density control reduces computational effort by removing non-essential landmarks from the map. A measure of *feature visibility* is introduced as a metric for deciding which features to keep and which to delete. That is, a feature has greater utility for SLAM if it is observable from many viewpoints than, say, a more stable feature visible only from a restricted vantage.

Features become obsolete due to changes in the structure of dynamic environments and the basic criterion for removal occurs when a feature is predicted visible but is not observed. A practical definition of “predicted visibility” is overlooked in the literature, and this thesis presents a criterion specifically applicable to a laser sensor.

Contributions

- Presentation of motivations for feature management: to prevent excessive clutter accumulation and to allow the map to adapt to structural change.
- Development of a mechanism for feature density control (a form of clutter reduction), based on a feature visibility metric.
- Development of a laser scan specific method for the removal of obsolete features.

7.1.4 Network Coupled Feature Maps

NCFM is a global map management framework that divides the world into a mosaic of independent local submaps connected by a network of relative pose estimates. Standard (optimal) stochastic SLAM is performed at a local level and a consistent suboptimal SLAM is performed at a global level. NCFM is computationally cheap and mathematically stable, not susceptible to large-scale non-linearity problems. Furthermore, it converges at a global level, and is conjectured to achieve a near optimal lower bound.

The ability of NCFM to address loop closure is one of its major benefits, and a complete strategy for cycle detection is presented. Cycle detection is realised by an efficient first-pass search, a second-pass detection of possible cycles, and finally, loop confirmation and the insertion of submap couplings.

Contributions

- NCFM effectively solves the problems of efficiency, convergence and non-linearity for large-scale SLAM.
- NCFM facilitates efficient and reliable cycle detection; it provides a solution to the loop closure problem.

7.2 Future Research

This section proposes several areas of this research requiring completion and extension. Furthermore, the contributions in this thesis give scope to numerous applications worthy of future investigation, and some promising directions are suggested below.

7.2.1 CCDA Algorithm Extensions

The following extensions are proposed for the CCDA algorithm.

- Implement a version of maximum clique search that stores multiple large cliques. This enables testing of various ambiguity management methods such as MHT with the k largest cliques.
- Implement alternative clique search algorithms (as presented in the literature) and compare their efficiency with the simple randomised search.
- Investigate environmental influences on batch association reliability and minimum acceptable batch size (i.e., data association “confidence”, as discussed in Section 3.5.2).
- Compare the tracking version of CCDA to JCBB [106] in terms of search efficiency and constraint action. Test whether a set of associations accepted by CCDA is also accepted by JCBB for the same χ^2 acceptance probability (and vice versa).
- Apply CCDA to different feature types and sensing modalities (esp. vision).

7.2.2 Scan Correlation Applications

Two applications for scan correlation are proposed here. First, in a mining tunnel environment, a precise metric location estimate is not particularly advantageous, and this kind of environment is suited to a simple topological-metric map as follows. The corridor intersections are described by a scan of unprocessed data (or, perhaps a small number of aligned scans), representing the map nodes. A weaker metric estimate is required between nodes, and the vehicle pose may be estimated via correlation-based dead reckoning. Place recognition is then performed by scan correlation with the node template scans.

A second application for scan correlation is stochastic SLAM in environments devoid of geometric features. The key idea is to describe portions of the environment by clusters of unprocessed data. That is, a scan of data is segmented into clusters representing objects or distinctive “regions of interest,” and each cluster serves as a static model, or template, for the object it describes. For each object, the representative data is referenced according to a local coordinate frame, and the global pose of the local frame $\mathbf{x}_f = (x_f, y_f, \phi_f)$ defines the object’s global location. Therefore, stochastic SLAM can be performed as follows. The object coordinate frame poses are stored in the augmented state vector $\mathbf{x}_a = [\mathbf{x}_v^T, \mathbf{x}_{f_1}^T, \dots, \mathbf{x}_{f_n}^T]^T$, and each time an object is observed, a likelihood function of its pose relative to the vehicle is obtained via scan correlation with its template data set. A Gaussian approximation of this likelihood function is then used to perform an EKF update of the state estimate. As new objects are observed, new static models are stored and new reference frames are added to the state. (Note, an object template might be produced from a conglomerate of aligned data clusters from the first few scans of the object.) The essential advantage of this application is that it permits EKF-based SLAM based on point-locations without requiring geometric feature models.

7.2.3 Feature Management Extensions

Some future directions for feature management are as follows.

- Implement a version of constrained initialisation for bearing-only or range-only SLAM. Here, the primary question is: does this method translate to sensors where a point feature location is not fully defined by a single measurement? (Note, significant work in this area has been presented in [91].)
- Improve the robustness of constrained initialisation by employing batch association to test all tentative features at once, and obtain a mutually compatible subset.
- Develop a more accurate and complete utility measure for density control. This would replace the current, rather crude, metric of “feature visibility.”
- Define more general criteria for the “predicted visibility” of features for obsolete feature removal. This problem may turn out to be application specific (e.g., models of occlusion, different sensing properties, etc).
- Implement “partial SLAM,” where all landmarks outside the current field-of-view are removed, and incorporate GPS information to obtain a fused global estimate.

7.2.4 NCFM SLAM Extensions

Most of the interesting properties of NCFM have not yet been implemented. A full experimental verification of NCFM SLAM is proposed as essential future work. This includes: local submap SLAM, vehicle transition, submap creation, coupling updates, and cycle detection.

Furthermore, the global consistency and convergence properties of NCFM require theoretical confirmation. This involves formal proof of the claims: (i) that treating correlated edge couplings as though they were independent is consistent (and conservative), and (ii) that global convergence approaches the same lower limit as full SLAM.

Future work may also investigate the following extensions to NCFM SLAM. First, to determine whether the conservative covariance estimates obtained from edge coupling summation may be improved given that the uncertainty of intermediate submap origins is irrelevant to a global estimate. And second, to investigate how absolute location information, such as GPS, might be incorporated into the map update.

7.2.5 Longer Term Developments

The original stochastic SLAM algorithm introduced an uncertainty measure that enabled consistent map building. With subsequent extensions, including those presented in this thesis, such as techniques for robust data association and tractable computation, stochastic SLAM is now possible at a very large scale.

Nevertheless, robust autonomous navigation in unstructured outdoor environments is far from a solved problem, and some difficult open problems remain. These fall into two main categories.

The first category is process modelling. Outdoor SLAM needs to contend with $2\frac{1}{2}$ -D and 3-D environments, and rugged terrain. Thus, it is necessary to develop process models that represent high-speed, non-planar motion and high levels of slip. One possible approach is to combine kinematic (i.e., encoder-based) models with INS and external-sensor based dead reckoning methods.

The second category is observation modelling. Scanning range lasers are not well suited to non-planar environments (although, given a non-planar motion model, improved results are possible), and alternative sensing and modelling methods are likely to be more effective. Using sensors better suited to 3-D (e.g., pan-tilt scanning laser, radar, vision), the key problem is to develop appropriate probabilistic observation models for them. For any given sensor, this problem includes the development of generic feature models that permit real-time data interpretation, calculation of egomotion and SLAM. Ideally, this would mean devising a feasible general-purpose landmark model that does require tailoring to new environments and applications.

The problem of sensor and landmark modelling, particularly for complex sensor information like vision, is by far the most difficult challenge for the implementation of general-purpose SLAM.

Appendix A

Environments

This appendix describes the three environments used for experimental evaluation of the techniques described in this thesis. Each description contains a photograph of a typical portion of the environment and an image produced from laser data obtained during experimental trials. The laser-based images were generated by first estimating the global pose of the laser for each scan (e.g., using SLAM) and then accumulating the unprocessed scan points into a single global scene. These images give a good impression of the types of objects visible to the laser sensor, both static and dynamic.

The particular characteristics of each environment that would influence the experimental results are discussed. These include the predominant types of static features and dynamic objects, and the quality of the road surface and surrounding terrain. Also noted is the configuration of the laser, in terms of scan rate and maximum effective range, and the driving speed of the vehicle. The vehicle speed is included to convey an idea of dynamic effects, such as roll and pitch, which cause most of the difficulties in interpreting the laser data.

The first two environments are outdoor regions where the observable landmarks are almost exclusively tree trunks. This makes (point) feature extraction reasonably straight forward. Since this thesis is concerned mostly with data association and map management, this simplification is not considered important and alternative feature models for other environments are common in the literature (particularly for indoors).

A.1 Urban Parkland

The park environment (see Figure A.1) consists of gently undulating terrain bounded along one side by a major highway. Much of the region contains trees which provides the main source of static features for localisation. The road surface is mostly smooth bitumen, although some parts are in disrepair and some of the vehicle trajectory is off-road.

There are three sources of dynamic objects in this environment. The first is the traffic moving along the adjacent highway, which at times occupies a large portion of the laser field-of-view. Second, there are a small number of people walking through the park during testing (although this is a very minor factor). Finally, and most importantly, laser tilt—caused by vehicle acceleration and irregularities in the ground surface—sometimes results in

large portions of a scan being corrupted by ground returns. These transient measurements both occlude good information and give an illusion of moving landscape as the laser cross-section shifts with subsequent scans. Embankments and other sloping structures have a similar effect.

The configuration of the scanning laser and indicative vehicle speeds for the trials in this environment are as follows.

<i>Laser Specifications</i>
Scan rate: 4.7 Hz
Maximum effective range: ~ 40 m

<i>Vehicle Speed</i>
Maximum speed: 5.3 m/s (19 km/hr)
Average speed: 2.8 m/s (10 km/hr)
Maximum turn rate: 0.6 rad/s (34 deg/s)



Figure A.1: Park environment.

A laser-based image for this environment is shown in Figure A.2. The trunks of trees are easily discernible as small discrete circles, while larger more blurred patches indicate shrubs (e.g., see coordinates $(-46, 5)$ and $(35, -30)$).

There are a lot of smeared portions in this picture that are caused by the motion of dynamic objects. The entire bottom section (from about -35 downward in the y-axis) is due to moving traffic along the adjacent highway. The set of dots forming an approximate line from $(-48, -3)$ to $(-28, -2)$ was caused by a person walking across the laser field-of-view over successive scans. Another fainter person-derived line is shown between $(-31, 1)$ and $(-11, 1)$. The remaining source of smearing is ground sweeps with the major occurrences in this picture shown about the regions $(-5, 5)$ and $(95, 5)$. The shifting nature of ground sweeps is most evident in the latter of these two examples. Other ground sweep interference occurred during the test run, some more severe than those shown here.

Note that in the upper-middle section of the picture, no stable features are visible. When the semicircle of the laser field-of-view was directed into this region (usually only for short periods), data association was obviously impossible and vehicle localisation was forced to rely solely on the inertial dynamic model. Other regions of sparse feature density exist in the areas right and left of the region pictured. In these regions, more accurate and reliable results would be possible with the incorporation of encoder-based odometry.

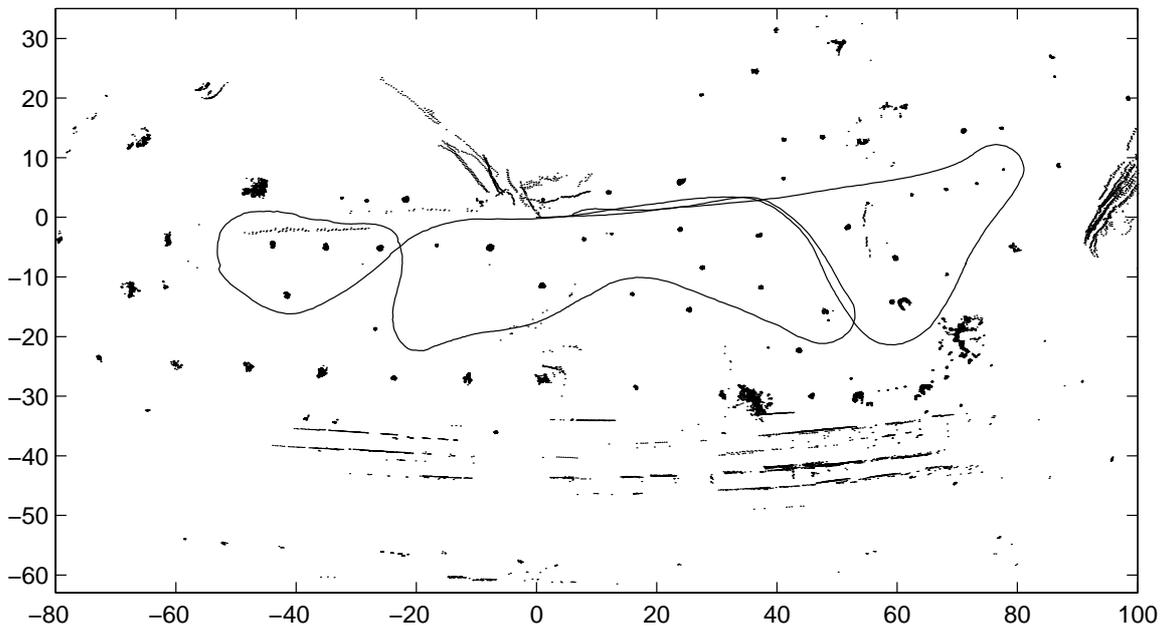


Figure A.2: Laser image of park environment. This image depicts all the laser returns accumulated over the test trajectory shown.

A.2 Internal Road of a Country Resort

The internal road environment is essentially static with ground sweeps from embankments in the terrain (and, to a lesser degree, laser tilt) being the only significant source of dynamic information. As with the park environment, the main source of features here are tree trunks. However, as is evident in Figure A.3, the density of trees is much greater; making data association a decidedly greater challenge.

The road is smooth bitumen, and the terrain is either flat or gently sloping, so that generally the laser scan information is close to ideal. However, there is one section of the trial run that is off-road over rough terrain and another where, for a short period, no visible features are available within the laser field-of-view. The off-road section, over which the vehicle traverses a small loop, is further encumbered by the presence of a large embankment and few stable features, and localisation from laser-only information becomes rather unreliable (see Section 5.4 for more details). However, in the other regions, where features are plentiful, the data association methods presented in this thesis perform extremely well.

<i>Laser Specifications</i>
Scan rate: 4.7 Hz
Maximum effective range: ~ 50 m

<i>Vehicle Speed</i>
Maximum speed: 9.2 m/s (33 km/hr)
Average speed: 4.7 m/s (17 km/hr)
Maximum turn rate: 0.6 rad/s (34 deg/s)



Figure A.3: Internal road environment.

The laser image shown in Figure A.4 depicts a region densely covered by trees. In this region, batch data association, as presented in Chapter 3, is absolutely essential as individual associations based on vehicle pose would quickly fail. The trunks of trees are easily discernible as small discrete points in the image, while shrubs and undergrowth appear as clumps of semi-merged points. The rectangular objects in the scene are small cabins. The clean straight lines defining these cabins indicate the accuracy of the SLAM algorithm used to generate this image.

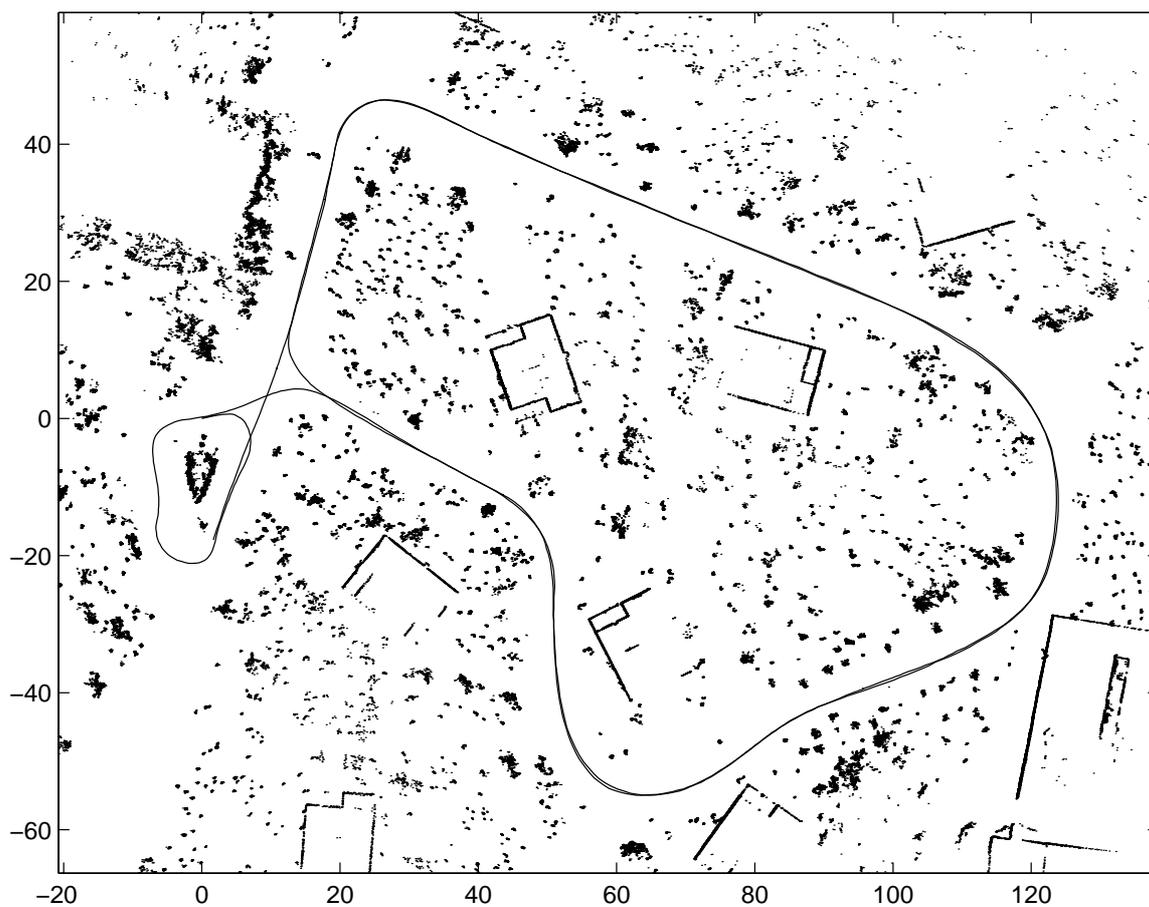


Figure A.4: Laser image of internal road environment.

A.3 Underground Mine Tunnel

The underground mine tunnel is substantially different to the previous two environment in that it possesses no discrete features but is characterised entirely by the texture of its walls (see Figure A.5). This environment is completely static with no dynamic objects present, and the road surface is flat and smooth. However, some variation in the perceived wall texture can occur due to the roll of the vehicle at higher speeds—which shifts the laser scan cross-section.

Extraction of stable parametric features in this environment is extremely difficult, although some work has been done to find points of maximum curvature using scale-space methods [94]. On the other hand, the quantity of texture in the wall surface permits very accurate and reliable scan correlation, which enables accurate laser-based dead reckoning, and is particularly tenable to place recognition and a topological-metric implementation.

<i>Laser Specifications</i>
Scan rate: 2.3 Hz
Maximum effective range: ~ 20 m

<i>Vehicle Speed</i>
Maximum speed: 4.5 m/s (16 km/hr)
Average speed: 3.2 m/s (12 km/hr)
Maximum turn rate: 0.6 rad/s (34 deg/s)

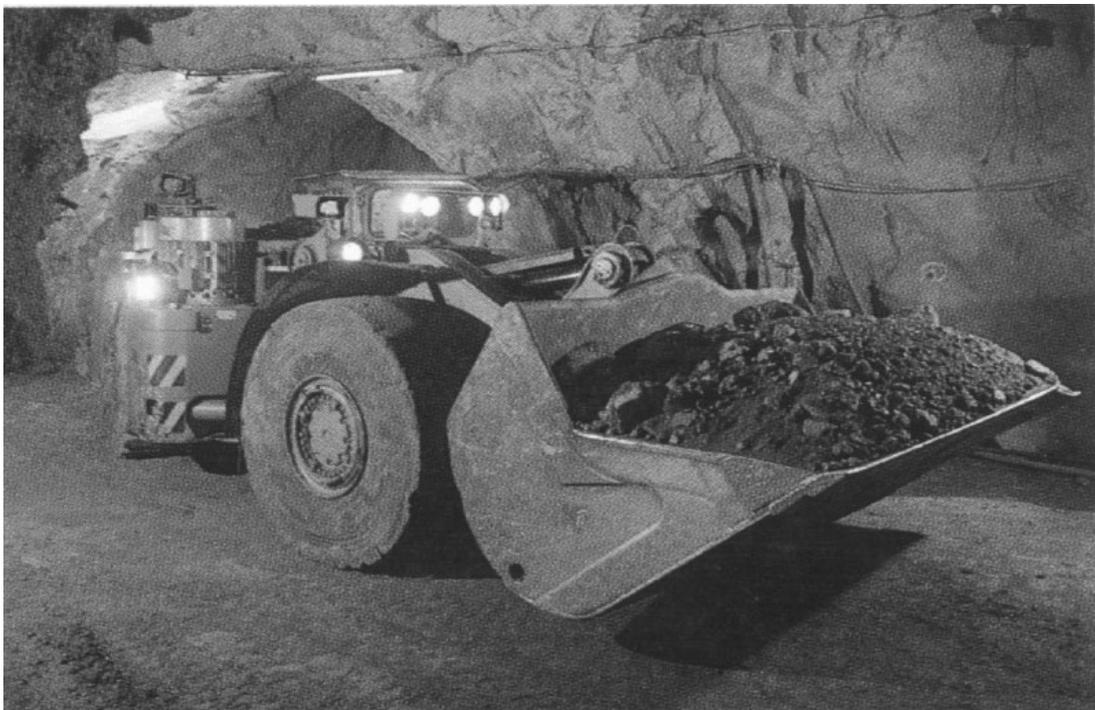


Figure A.5: Mine tunnel environment. Note the smooth road surface and distinctive wall texture. Coal is extracted from this mine using LHD (load, haul, dump) vehicles such as the one pictured in the foreground.

Experiments in the mine were performed using a diesel 4WD vehicle with the laser attached to the front bumper. The laser image shown in Figure A.6 is compiled using scan correlation based dead reckoning (i.e., the pose of the laser for each scan is found relative to the laser pose for the previous scan using the maximum likelihood scan correlation method presented in Chapter 4). The clarity and definition of the walls in the image demonstrate the accuracy of this dead reckoning procedure. Note, in the bottom right-hand corner of the image, the vehicle performs a 3-point turn—accounting for the apparent discontinuity in the vehicle trajectory.

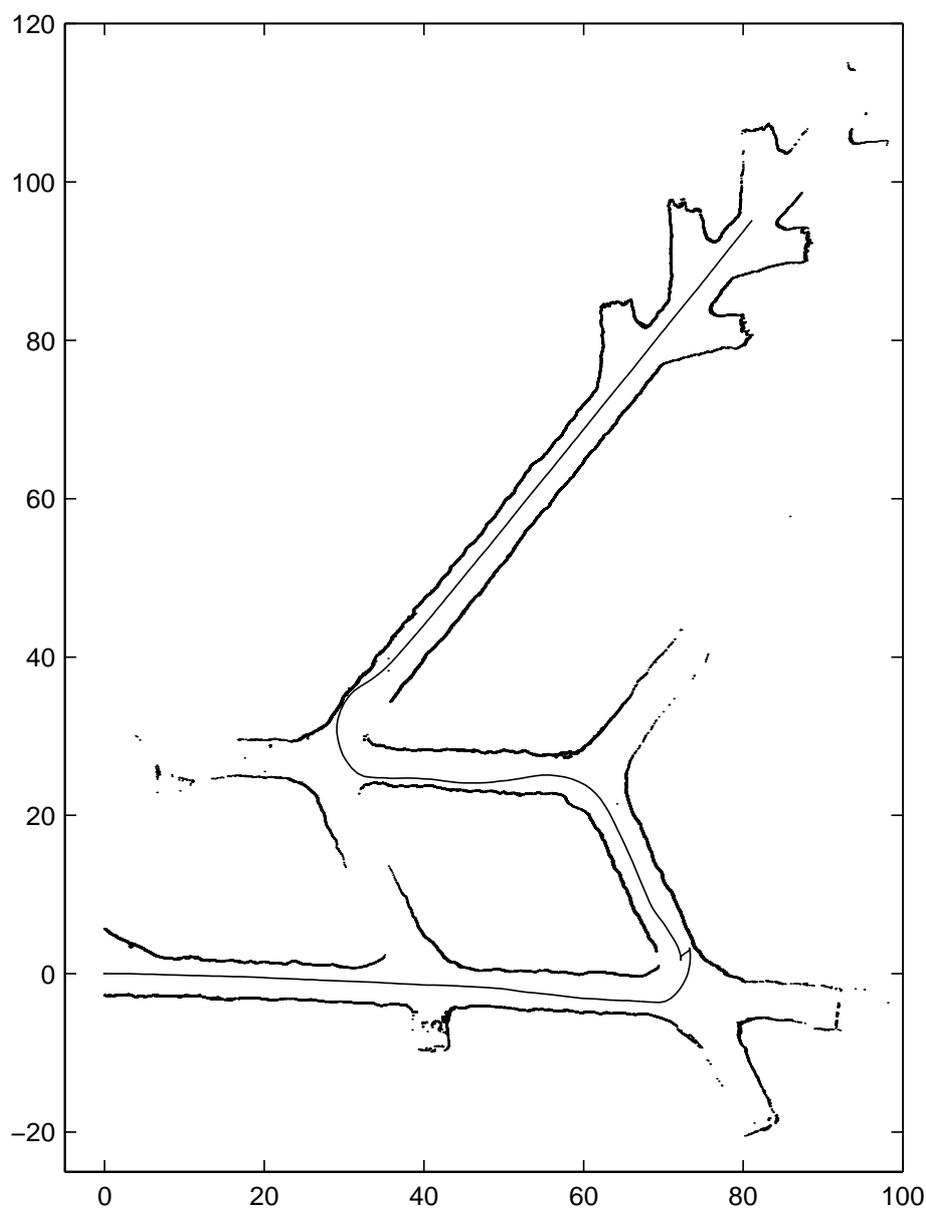


Figure A.6: Laser image of mine tunnel.

Appendix B

Kalman Filter Topics

This appendix contains a brief introduction to state-space representations and describes the basic linear Kalman filter and extended Kalman filter algorithms. Further information regarding Kalman filters can be found in the following texts [97, 6, 61].

B.1 State-Space

State space concerns the representation of a system in terms of a set of scalar parameters, or state variables, such that the system is defined by a state vector $\mathbf{x} = [x_1, \dots, x_n]^T$. Usually the exact condition of the state vector is not known, and there is a degree of uncertainty to each of the state variables. If the uncertainty has Gaussian distribution, then each variable can be defined in terms of its mean \hat{x}_i and standard deviation σ_i . Generally, the magnitude of uncertainty is given by the standard deviation squared (or variance) σ_i^2 , as in this form the uncertainties of independent variables can be added. For example, if x_1 and x_2 are independent scalar variables with standard deviations σ_1 and σ_2 respectively, and $x_3 = x_1 + x_2$ then $\sigma_3^2 = \sigma_1^2 + \sigma_2^2$. Therefore, the system \mathbf{x} is represented by its mean vector and covariance matrix.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix} \triangleq E[\mathbf{x}]$$
$$\mathbf{P}_{\mathbf{x}} = \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{1n}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{1n}^2 & \cdots & \sigma_{nn}^2 \end{bmatrix} \triangleq E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T]$$

where $E[\mathbf{x}]$ defines the *expected value* or *mean* as follows.¹

$$E[\mathbf{x}] \triangleq \int_{-\infty}^{\infty} \mathbf{x} p(\mathbf{x}) d\mathbf{x}$$

The off-diagonal terms of $\mathbf{P}_{\mathbf{x}}$ are called cross-correlations and represent the dependence of the uncertainty of one variable upon another.

¹The function $p(\mathbf{x})$ defines the probability density of \mathbf{x} . More information about probability density functions (PDFs) is provided in Chapter 4 and Appendix D.

B.1.1 Linear Transformations

If \mathbf{x} is transformed to \mathbf{y} by a system of linear equations \mathbf{F} , then the covariance matrix is transformed as follows.

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{F}\hat{\mathbf{x}} \\ \mathbf{P}_y &= \mathbf{F}\mathbf{P}_x\mathbf{F}^T\end{aligned}$$

The addition of two independent state vectors is similar.

$$\begin{aligned}\hat{\mathbf{x}}_3 &= \mathbf{A}\hat{\mathbf{x}}_1 + \mathbf{B}\hat{\mathbf{x}}_2 \\ \mathbf{P}_3 &= \mathbf{A}\mathbf{P}_1\mathbf{A}^T + \mathbf{B}\mathbf{P}_2\mathbf{B}^T\end{aligned}$$

Non-linear transformations, however, present a problem as it is no longer possible to simply transform the covariance matrix from one space to the other. The usual solution to this problem is to approximate the covariance transform by linearising the transformation functions. Given a set of non-linear functions \mathbf{f} , the linearised Jacobian matrix $\nabla\mathbf{f}_x$ is defined as the partial derivative of \mathbf{f} with respect to \mathbf{x} about the point $\hat{\mathbf{x}}$.

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{bmatrix}$$

$$\nabla\mathbf{f}_x = \left. \frac{\partial\mathbf{f}}{\partial\mathbf{x}} \right|_{\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}_{\hat{\mathbf{x}}}$$

Assuming that \mathbf{f} is reasonably linear over a small region, the linearised transformation matrix approximates \mathbf{f} for values of \mathbf{x} close to $\hat{\mathbf{x}}$. Therefore, the covariance transformation can be calculated as for the linear case.

$$\mathbf{P}_y = \nabla\mathbf{f}_x\mathbf{P}_x\nabla\mathbf{f}_x^T$$

A problem with the linearised approach is that the transformations introduce a bias into the covariance estimate, and may incur a significant error for very non-linear functions. An alternative method that addresses this issue is the *unscented transformation* [76, 75] which transforms a set of points through the non-linear function and reconstructs a Gaussian distribution to encompass these points.

Example B.1

Conversion between polar and Cartesian coordinate systems. A polar measurement \mathbf{p} , such as returned by a range-bearing laser, is often required as a point \mathbf{x} in Cartesian space.

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{r} \\ \hat{\theta} \end{bmatrix} \xrightarrow{\mathbf{f}} \hat{\mathbf{x}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}$$

$$\mathbf{P}_p = \begin{bmatrix} \sigma_{rr}^2 & \sigma_{r\theta}^2 \\ \sigma_{r\theta}^2 & \sigma_{\theta\theta}^2 \end{bmatrix} \xrightarrow{\mathbf{f}} \mathbf{P}_x$$

where the non-linear transform \mathbf{f} is given by

$$\mathbf{x} = \mathbf{f}(\mathbf{p}) = \begin{bmatrix} r \cos \theta \\ r \sin \theta \end{bmatrix}$$

The transformed estimate $\hat{\mathbf{x}}$ can be calculated directly from the non-linear function but the covariance must be determined from the Jacobian.

$$\begin{aligned} \nabla_{\mathbf{f}_p} &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \right|_{\hat{\mathbf{p}}} = \begin{bmatrix} \cos \hat{\theta} & -\hat{r} \sin \hat{\theta} \\ \sin \hat{\theta} & \hat{r} \cos \hat{\theta} \end{bmatrix} \\ \mathbf{P}_x &= \nabla_{\mathbf{f}_p} \mathbf{P}_p \nabla_{\mathbf{f}_p}^T \end{aligned}$$

The inverse transformation from Cartesian to polar coordinates is similar. The function and its Jacobian are as follows.

$$\begin{aligned} \mathbf{p} = \mathbf{h}(\mathbf{x}) &= \begin{bmatrix} \sqrt{x^2 + y^2} \\ \arctan\left(\frac{y}{x}\right) \end{bmatrix} \\ \nabla_{\mathbf{h}_x} &= \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\hat{x}}{\sqrt{\hat{x}^2 + \hat{y}^2}} & \frac{\hat{y}}{\sqrt{\hat{x}^2 + \hat{y}^2}} \\ \frac{-\hat{y}}{\hat{x}^2 + \hat{y}^2} & \frac{\hat{x}}{\hat{x}^2 + \hat{y}^2} \end{bmatrix} \end{aligned}$$

B.2 The Kalman Filter

The *Kalman filter* is an algorithm for fusing information in state-space. It is applicable to systems where both the process (state transitions) and observations of the state are linear and the associated uncertainties are Gaussian. Essentially, it works by a two step cycle, a prediction step and an update step, to recursively produce a state estimate with minimum mean squared error (i.e., minimise the trace of the state covariance matrix).

Given a system defined by the state \mathbf{x}_k (at time k) and modelled by the process

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{q}_k \quad (\text{B.1})$$

where \mathbf{u}_k represents some independent state vector (usually a control input) and \mathbf{q}_k denotes Gaussian white noise, the *predict* step of the Kalman filter operates as follows.

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\hat{\mathbf{u}}_k \quad (\text{B.2})$$

$$\mathbf{P}_{k+1}^- = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{B}\mathbf{U}_k\mathbf{B}^T + \mathbf{Q}_k \quad (\text{B.3})$$

The $-$ sign indicates that this is a prediction or an *a priori* state estimate. The matrices \mathbf{P}_k , \mathbf{U}_k and \mathbf{Q}_k are the covariance matrices for \mathbf{x}_k , \mathbf{u}_k and \mathbf{q}_k respectively. If, at time $k+1$, an independent observation is made of the state

$$\mathbf{z}_{k+1} = \mathbf{H}\mathbf{x}_{k+1} + \mathbf{r}_{k+1} \quad (\text{B.4})$$

where \mathbf{r}_{k+1} represents Gaussian white noise with covariance \mathbf{R}_{k+1} , then an improvement can be made to the state estimate. That is, the update estimate of \mathbf{x}_{k+1} will be a weighted sum

of the predicted state and the observed state. The Kalman filter determines the optimal weighting as follows.

$$\nu_{k+1} = \mathbf{z}_{k+1} - \mathbf{H}\hat{\mathbf{x}}_{k+1}^- \quad (\text{B.5})$$

$$\mathbf{S}_{k+1} = \mathbf{H}\mathbf{P}_{k+1}^- \mathbf{H}^T + \mathbf{R}_{k+1} \quad (\text{B.6})$$

$$\mathbf{W}_{k+1} = \mathbf{P}_{k+1}^- \mathbf{H}^T \mathbf{S}_{k+1}^{-1} \quad (\text{B.7})$$

The vector ν_{k+1} is the error (or residual) between the predicted and actual observation. This error is commonly termed the innovation and the matrix \mathbf{S}_{k+1} represents the innovation covariance. The optimal weighting \mathbf{W}_{k+1} is known as the Kalman gain. The updated state estimate (or *a posteriori* estimate) is then determined.

$$\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \mathbf{W}_{k+1}\nu_{k+1} \quad (\text{B.8})$$

$$\mathbf{P}_{k+1}^+ = \mathbf{P}_{k+1}^- - \mathbf{W}_{k+1}\mathbf{S}_{k+1}\mathbf{W}_{k+1}^T \quad (\text{B.9})$$

Notice that the Kalman filter estimates the covariance \mathbf{P}_{k+1} independent of the value of the state mean $\hat{\mathbf{x}}_{k+1}$ (i.e., the value of the mean estimate does not affect the covariance estimate). In fact, for linear systems, it is possible to compute the state covariance off-line.

B.3 The Extended Kalman Filter

Most real systems, and certainly the systems in this thesis, are not governed by linear equations, and the basic Kalman filter is insufficient for these tasks. To cater for non-linear estimation problems the *extended Kalman filter* (EKF) was devised. For example, a state transition is modelled by a non-linear function \mathbf{f} and an observation of the state modelled by a non-linear function \mathbf{h} as follows.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{q}_k \quad (\text{B.10})$$

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{r}_{k+1} \quad (\text{B.11})$$

The key attribute of the EKF is to linearise the functions \mathbf{f} and \mathbf{h} about the point of the state mean. Therefore, unlike the linear Kalman filter, the state covariance estimate is no longer independent of the state mean estimate as they are coupled by the Jacobian calculations. The prediction step becomes

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k) \quad (\text{B.12})$$

$$\mathbf{P}_{k+1}^- = \nabla \mathbf{f}_{\mathbf{x}_k} \mathbf{P}_k \nabla \mathbf{f}_{\mathbf{x}_k}^T + \nabla \mathbf{f}_{\mathbf{u}_k} \mathbf{U}_k \nabla \mathbf{f}_{\mathbf{u}_k}^T + \mathbf{Q}_k \quad (\text{B.13})$$

where the Jacobian $\nabla \mathbf{f}_{\mathbf{x}_k}$ is the partial derivative of \mathbf{f} with respect to \mathbf{x} at the point $(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k)$ and the Jacobian $\nabla \mathbf{f}_{\mathbf{u}_k}$ is similarly defined for \mathbf{u} .

$$\nabla \mathbf{f}_{\mathbf{x}_k} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} \right|_{(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k)} \quad (\text{B.14})$$

$$\nabla \mathbf{f}_{\mathbf{u}_k} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}_k} \right|_{(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k)} \quad (\text{B.15})$$

Calculation of the Kalman gain is performed similarly by linearising the observation function \mathbf{h} .

$$\nu_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1}^-) \quad (\text{B.16})$$

$$\mathbf{S}_{k+1} = \nabla \mathbf{h}_{\mathbf{x}_{k+1}} \mathbf{P}_{k+1}^- \nabla \mathbf{h}_{\mathbf{x}_{k+1}}^T + \mathbf{R}_k \quad (\text{B.17})$$

$$\mathbf{W}_{k+1} = \mathbf{P}_{k+1}^- \nabla \mathbf{h}_{\mathbf{x}_{k+1}}^T \mathbf{S}_{k+1}^{-1} \quad (\text{B.18})$$

where the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_{k+1}}$ is given by

$$\nabla \mathbf{h}_{\mathbf{x}_{k+1}} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_{k+1}} \right|_{\hat{\mathbf{x}}_{k+1}^-} \quad (\text{B.19})$$

The state update is then calculated using the normal Kalman update equations B.8 and B.9. The EKF operates on the assumption that the functions \mathbf{f} and \mathbf{h} are near-linear so that the linearised transformations sufficiently approximate the correct covariance transformations. However, for highly non-linear functions, linearisation may generate inconsistent uncertainty estimates and a method like the unscented transform might provide more accurate results. Also, as with the standard linear Kalman filter, the EKF assumes Gaussian uncertainty distributions for all estimates and large deviations from this ideal may also lead to unreliable performance.

Appendix C

Insert and Observe State Augmentation

An alternative to deriving specialised Jacobians for the initialisation of state variables is the method of *insert and observe*, which enables state element initialisation via the EKF update equations. The method is appropriate for any state augmentation procedure where new elements are added to an existing state vector. Initialisation is performed by first setting the means of the inserted elements to arbitrary values and their covariance diagonals to infinity (i.e., total uncertainty).

$$\hat{\mathbf{x}}_{aug} = \begin{bmatrix} \hat{\mathbf{x}}_{old} \\ \hat{\mathbf{x}}_{new} \end{bmatrix} \quad (\text{C.1})$$

$$\mathbf{P}_{aug} = \begin{bmatrix} \mathbf{P}_{old} & \mathbf{0} \\ \mathbf{0} & \infty \mathbf{I}_{new} \end{bmatrix} \quad (\text{C.2})$$

Obtaining the correct mean and covariance values for the added elements (including cross-correlations to the old portion of the state vector) is then simply a matter of applying the normal observation update equations.

In practice, the added mean elements are initially set to values expected by the new observation information (i.e., a close initial guess); this prevents linearisation problems during the EKF update. The initial covariance diagonals are set to a suitably large value (e.g., $\sigma^2 = 10^{12}$), as an approximation to infinity. While using non-infinite variances actually constitutes a reduction in uncertainty without the addition of information, this inconsistency is extremely small and easily absorbed by the added process and observation stabilising noise.

C.1 New Feature Initialisation

This example of the insert and observe method is an alternative to the map augmentation method presented in Section 2.2.4. However, the following approach is not recommended for feature initialisation in practice, since the direct Jacobian-based method is both more exact and more efficient.

Given the observation $\mathbf{z} = [r, \theta]^T$ of a new feature (x_i, y_i) where the observation is modelled as

$$\hat{\mathbf{z}}_i = \mathbf{h}_i(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_i) = \begin{bmatrix} \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_i - \hat{y}_v}{\hat{x}_i - \hat{x}_v}\right) - \hat{\phi}_v \end{bmatrix} \quad (\text{C.3})$$

the feature is added to the state vector \mathbf{x}_a with approximately¹ infinite uncertainty.

$$\hat{\mathbf{x}}_{aug} = \begin{bmatrix} \hat{\mathbf{x}}_a \\ x_v + r \cos(\theta + \phi_v) \\ y_v + r \sin(\theta + \phi_v) \end{bmatrix} \quad (\text{C.4})$$

$$\mathbf{P}_{aug} = \begin{bmatrix} \mathbf{P}_a & \mathbf{0} \\ \mathbf{0} & \alpha \mathbf{I} \end{bmatrix} \quad (\text{C.5})$$

The posterior augmented state estimate can then be obtained via the normal SLAM update equations (i.e., Equations 2.17 to 2.22 from Section 2.2.3).

In the event of long term SLAM, where a huge number of features are initialised by this method, it is conceivable that the small decrease in vehicle pose uncertainty due to finite α may accumulate to the point where the estimate becomes inconsistent. To avoid this problem, a simple modification to the update equations (i.e., Equations 2.21 and 2.22) may be implemented so that only the new feature portion of the state vector, and only the new feature block and its off-diagonals of the state covariance matrix, are changed. With this modification, the information increase is confined to the new feature only and does not accumulate.

C.2 Vehicle Pose Initialisation in a Partially Known Map

The insert and observe initialisation method is most advantageous when used to perform an augmentation of the state vector that requires multiple simultaneous observations (i.e., a batch observation). Again, the elements to be initialised are simply given approximately correct nominal values and large uncertainty, and are updated to the correct values using the observation information.

If a SLAM map has been constructed for a particular environment, the vehicle states $\hat{\mathbf{x}}_v$ can be removed from the map in a consistent manner by simply deleting the vehicle elements from the state vector and deleting the vehicle-correlated rows and columns from the covariance matrix. This enables the map $\hat{\mathbf{x}}_m$ to be decoupled from the vehicle pose estimate.

Supposing the vehicle (or another vehicle) later observes a set of features from the map using the non-tracking CCDA algorithm (see Section 3.3.3), and it is desired to append the vehicle pose to the map and continue SLAM map building. An initial guess of the vehicle pose \mathbf{x}_{guess} can be obtained using a batch relative pose estimation technique—such as the closed-form least-squares algorithm presented in [73, 93]. The augmented state vector is

¹A practical approximation to infinity is provided by a large scalar α .

then constructed as follows.

$$\hat{\mathbf{x}}_a^- = \begin{bmatrix} \hat{\mathbf{x}}_{guess} \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (\text{C.6})$$

$$\mathbf{P}_a^- = \begin{bmatrix} \alpha \mathbf{I}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_m \end{bmatrix} \quad (\text{C.7})$$

The batch observation information is subsequently used to update this state estimate to the proper value (i.e., via Equations 2.17 to 2.22).

$$\hat{\mathbf{x}}_a^+ = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (\text{C.8})$$

$$\mathbf{P}_a^+ = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_m \end{bmatrix} \quad (\text{C.9})$$

C.3 Relative Pose Estimation

The following technique is a variation on the vehicle pose initialisation example which enables calculation of the relative pose (and uncertainty) between two scans of feature points.

To begin, the state vector is constructed with the vehicle at pose $\hat{\mathbf{x}}_v = [0, 0, 0]^T$, with zero uncertainty, and an initial map composed of the features from the first scan.²

$$\hat{\mathbf{x}}_a = \begin{bmatrix} \hat{\mathbf{0}}_v \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (\text{C.10})$$

$$\mathbf{P}_a = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_m \end{bmatrix} \quad (\text{C.11})$$

Notice that each feature $\hat{\mathbf{x}}_i = [\hat{x}_i, \hat{y}_i]^T$ in the map is not correlated to the vehicle or to any of the other features, and is derived from the measurement $\mathbf{z}_i = [r_i, \theta_i]^T$ as follows.

$$\hat{\mathbf{x}}_i = \mathbf{g}(\mathbf{z}_i) = \begin{bmatrix} r_i \cos \theta_i \\ r_i \sin \theta_i \end{bmatrix} \quad (\text{C.12})$$

$$\mathbf{P}_i = \nabla \mathbf{g}_z \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \nabla \mathbf{g}_z^T \quad (\text{C.13})$$

where the Jacobian $\nabla \mathbf{g}_z$ is given by

$$\nabla \mathbf{g}_z = \frac{\partial \mathbf{g}}{\partial \mathbf{z}_i} = \begin{bmatrix} \cos \theta_i & -r_i \sin \theta_i \\ \sin \theta_i & r_i \cos \theta_i \end{bmatrix} \quad (\text{C.14})$$

The next step is to guess the vehicle pose for the second scan. If there is no *a priori* information, this guess might be obtained by first finding the set of feature associations between the two scans via the non-tracking version of the CCDA algorithm (see Section 3.3.3).

²Equations C.10 and C.11 are also used for initialising the state vector for SLAM.

Then, as in Section C.2, a guess of the relative pose can be obtained using a simple batch estimation algorithm, and the pose covariance diagonals are assigned a large uncertainty.

$$\hat{\mathbf{x}}_a^- = \begin{bmatrix} \hat{\mathbf{x}}_{guess} \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (\text{C.15})$$

$$\mathbf{P}_a^- = \begin{bmatrix} \alpha \mathbf{I}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_m \end{bmatrix} \quad (\text{C.16})$$

On the other hand, if there exists *a priori* information about the relative pose (e.g., from odometry or dynamic constraints), then the initial guess and uncertainty would be as follows.

$$\hat{\mathbf{x}}_a^- = \begin{bmatrix} \hat{\mathbf{x}}_{prior} \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (\text{C.17})$$

$$\mathbf{P}_a^- = \begin{bmatrix} \mathbf{P}_{prior} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_m \end{bmatrix} \quad (\text{C.18})$$

With this prior estimate, the tracking version of the CCDA algorithm (see Section 3.3.4) can be applied subsequently—making use of the absolute constraint information.

Having obtained a set of associations between the two scans, a full SLAM update is performed using Equations 2.17 to 2.22.

$$\hat{\mathbf{x}}_a^+ = \begin{bmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{x}}_m \end{bmatrix} \quad (\text{C.19})$$

$$\mathbf{P}_a^+ = \begin{bmatrix} \mathbf{P}_v & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_m \end{bmatrix} \quad (\text{C.20})$$

This produces a state estimate where the vehicle and map portions are correlated. If this method is used to obtain the relative pose between two isolated scans, then the vehicle pose estimate $\hat{\mathbf{x}}_v$ and \mathbf{P}_v can be extracted directly. However, if this method is used to calculate the change-in-pose over a sequence of scans (e.g., laser-based dead reckoning), then the map correlations become an issue, and direct use of \mathbf{P}_v may be inconsistent (see the example below).

Example C.1

Laser-based dead reckoning. Given a sequence of laser scans, the relative pose between the k th and $k+1$ th scan can be calculated using the method shown above. A naive dead reckoning implementation would make the change-in-pose estimate, $\hat{\mathbf{x}}_\delta$ and \mathbf{P}_δ , equal to the extracted relative pose estimate $\hat{\mathbf{x}}_v$ and \mathbf{P}_v .

Let the dead reckoning estimate for the pose of the k th scan, be denoted $\mathbf{x}_k = [x_k, y_k, \phi_k]^T$; the dead reckoning equations for the $k+1$ th scan are as follows.³

$$\hat{\mathbf{x}}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_\delta) = \begin{bmatrix} \hat{x}_k + \hat{x}_\delta \cos \hat{\phi}_k - \hat{y}_\delta \sin \hat{\phi}_k \\ \hat{y}_k + \hat{x}_\delta \sin \hat{\phi}_k + \hat{y}_\delta \cos \hat{\phi}_k \\ \hat{\phi}_k + \hat{\phi}_\delta \end{bmatrix}$$

$$\mathbf{P}_{k+1} = \nabla \mathbf{f}_{\mathbf{x}_k} \mathbf{P}_k \nabla \mathbf{f}_{\mathbf{x}_k}^T + \nabla \mathbf{f}_{\mathbf{x}_\delta} \mathbf{P}_\delta \nabla \mathbf{f}_{\mathbf{x}_\delta}^T$$

³These equations are a simple variation of the state prediction equations in Section 2.2.2 where the map portion of the state vector is empty.

where the Jacobians $\nabla \mathbf{f}_{\mathbf{x}_k}$ and $\nabla \mathbf{f}_{\mathbf{x}_\delta}$ are defined as

$$\nabla \mathbf{f}_{\mathbf{x}_k} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} \right|_{(\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_\delta)} = \begin{bmatrix} 1 & 0 & -\hat{x}_\delta \sin \hat{\phi}_k - \hat{y}_\delta \cos \hat{\phi}_k \\ 0 & 1 & \hat{x}_\delta \cos \hat{\phi}_k - \hat{y}_\delta \sin \hat{\phi}_k \\ 0 & 0 & 1 \end{bmatrix}$$

$$\nabla \mathbf{f}_{\mathbf{x}_\delta} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_\delta} \right|_{(\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_\delta)} = \begin{bmatrix} \cos \hat{\phi}_k & -\sin \hat{\phi}_k & 0 \\ \sin \hat{\phi}_k & \cos \hat{\phi}_k & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The problem with assigning $\mathbf{P}_\delta = \mathbf{P}_v$ is that it may result in an inconsistent estimate of \mathbf{P}_{k+1} because the relative pose estimates are pair-wise correlated (i.e., the change-in-pose estimate between scans $k-1$ and k is correlated to the change-in-pose estimate between scans k and $k+1$). In other words, the information in each scan is used twice, once with the preceding scan and once with the subsequent scan. The measure used in this thesis to prevent inconsistent dead reckoning is to expand the change-in-pose uncertainty by two (i.e., $\mathbf{P}_\delta = 2\mathbf{P}_v$), which caters for the worst case condition where the two adjacent change-in-pose estimates are fully correlated.⁴

⁴Ignoring correlations does not necessarily imply an inconsistent estimate and, in the particular case of sensor-based dead reckoning, the correlations actually tend to produce a self-correcting estimate. Thus, assigning $\mathbf{P}_\delta = \mathbf{P}_v$ is conservative in most, if not all, circumstances (see Section 6.2.6 for further explanation). In other words, while uncertainty inflation is necessary in general for functions of variables with unknown correlations, it is probably not required for the summation of change-in-pose estimates.

Appendix D

Particle Filters

This appendix introduces Bayesian (probabilistic) estimation and describes the particle filter algorithm as a practical method for implementing recursive Bayesian estimation for real systems. Particle filtering is an attractive estimation technique for systems with any of the following properties: (i) governed by non-linear transition models, (ii) observed via non-linear sensor models, and (iii) possessing arbitrary (non-Gaussian) uncertainty distributions.

D.1 Probability Density Functions

The probabilistic approach to estimation is based on the concept of the *probability density function* (PDF) which is used to define the uncertainty distribution of a set of random variables. In other words a PDF $p(\mathbf{x})$ expresses, for a particular random vector \mathbf{x} , the likelihood that the true state of \mathbf{x} lies within a particular region of the state-space \mathcal{X} . For the purposes of this thesis it is sufficient to understand the following PDF properties.¹

- A PDF $p(\mathbf{x})$ represents a functional mapping $p : \mathbf{x} \rightarrow \mathbb{R}$ for all $\mathbf{x} \in \mathcal{X}$.
- A PDF $p(\mathbf{x})$ is non-negative for all values of random vector \mathbf{x} .

$$p(\mathbf{x}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}$$

- The area (or volume) under a PDF is one.

$$\int_{-\infty}^{\infty} p(\mathbf{x}) d\mathbf{x} = 1$$

If there exist two random vectors \mathbf{x} and \mathbf{y} where the value of \mathbf{x} is to some degree dependent on the value of \mathbf{y} , then there exists a *conditional* PDF $p(\mathbf{x}|\mathbf{y})$. The conditional PDF $p(\mathbf{x}|\mathbf{y})$ may be understood as the probability or likelihood of \mathbf{x} given a fixed value of \mathbf{y} . If \mathbf{x} and \mathbf{y} are independent then $p(\mathbf{x}|\mathbf{y}) = p(\mathbf{x})$.

¹Excellent expositions of Bayesian estimation theory can be found in [97, 109, 125, 6].

D.2 Recursive Bayesian Estimation

This section gives a cursory introduction to Bayesian estimation² which involves tracking the PDF of a random vector over time given a model of its incremental change and a means of observing its actual state. A state vector \mathbf{x}_k at time k , is assumed to evolve over time as modelled by the state transition function \mathbf{f} .³

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{q}_k) \quad (\text{D.1})$$

where \mathbf{q}_k is an independent noise sequence and the PDF $p(\mathbf{q}_k)$ is known. An observation of the state is obtained at discrete intervals such that the measurement \mathbf{z}_k is modelled by the function \mathbf{h} .

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{r}_k) \quad (\text{D.2})$$

where \mathbf{r}_k is also an independent noise sequence with known PDF.

Assuming that at time k , the PDF $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ is known⁴ and an observation \mathbf{z}_k has been obtained, $p(\mathbf{x}_k|\mathbf{Z}_k)$ can be estimated by a two-step process. The first is a prediction step to find the prior PDF $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$.

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int_{-\infty}^{\infty} p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}) d\mathbf{x}_{k-1} \quad (\text{D.3})$$

where the $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is derived from the state transition model \mathbf{f} . It is assumed that the transition function is a Markov process such that $p(\mathbf{x}_k|\{\mathbf{x}_0, \dots, \mathbf{x}_{k-1}\})$ is equal to $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. That is, the PDF of \mathbf{x}_k is conditioned by \mathbf{x}_{k-1} only and is independent of all previous states.

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \int_{-\infty}^{\infty} \delta(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{q}_{k-1})) p(\mathbf{q}_{k-1}) d\mathbf{q}_{k-1} \quad (\text{D.4})$$

where $\delta(\cdot)$ is the Dirac delta function or impulse function (an infinitely brief pulse with infinite amplitude and unit area [19]). The second step, the update step, follows from Bayes theorem that the posterior PDF $p(\mathbf{x}_k|\mathbf{Z}_k)$ is given by

$$p(\mathbf{x}_k|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{\int_{-\infty}^{\infty} p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{k-1}) d\mathbf{x}_k} \quad (\text{D.5})$$

where the denominator of the equation is simply a scaling factor to normalise the area under the posterior to one. The PDF $p(\mathbf{z}_k|\mathbf{x}_k)$ is defined by the observation model \mathbf{h} .

$$p(\mathbf{z}_k|\mathbf{x}_k) = \int_{-\infty}^{\infty} \delta(\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k, \mathbf{r}_k)) p(\mathbf{r}_k) d\mathbf{r}_k \quad (\text{D.6})$$

For the case where the state transition and observation functions, \mathbf{f} and \mathbf{h} , are linear and the uncertainty distributions are all Gaussian, then the above equations are equivalent

²Sections D.2 and D.3 are based on [60] which provides an accessible introduction to the particle filter algorithm.

³The functions \mathbf{f} and \mathbf{h} may change over time and so should actually be written as \mathbf{f}_k and \mathbf{h}_k .

⁴The conditional PDF $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$ where \mathbf{Z}_{k-1} is the set $\{\mathbf{z}_1, \dots, \mathbf{z}_{k-1}\}$, can be interpreted as the PDF of \mathbf{x}_{k-1} given all the available observation information up to time $k-1$.

to the Kalman filter (see Appendix B for a description of the KF algorithm). If \mathbf{f} and \mathbf{h} are almost linear then, for Gaussian PDFs, Equations D.3 to D.6 may be adequately approximated by the extended Kalman filter. However, for general non-linear systems with arbitrary distributions, the EKF is unsuitable and it may be very difficult to find an analytical description of the posterior distribution. An alternative approach is to obtain an approximate estimate of the posterior PDF using samples. The technique of drawing state samples from the prior distribution and using these samples (in conjunction with state transition and observation information) to approximate the posterior is known as *particle filtering*.⁵

D.3 Particle Filtering: Sample Based Estimation

The basic particle filter algorithm is implemented as follows. First, assuming that the initial PDF $p(\mathbf{x}_0)$ is known, the algorithm is initialised by drawing n random samples $\{\mathbf{x}_0(1), \dots, \mathbf{x}_0(n)\}$ from this distribution. From this point on the conditional PDF of \mathbf{x} is represented entirely by samples and an analytical formulation of the distribution is not required. The evolution of $p(\mathbf{x}_k|\mathbf{Z}_k)$ is computed recursively using a three step process: prediction, likelihood weighting, and resampling.

1. Prediction is performed by passing each sample from the previous time step $\mathbf{x}_{k-1}(i)$ through the state transition function (Equation D.1).

$$\mathbf{x}_k^-(i) = \mathbf{f}(\mathbf{x}_{k-1}(i), \mathbf{q}_{k-1}(i)) \quad (\text{D.7})$$

where the sample $\mathbf{q}_{k-1}(i)$ is drawn⁶ from the known distribution $p(\mathbf{q}_{k-1})$. The resulting set of samples $\{\mathbf{x}_k^-(1), \dots, \mathbf{x}_k^-(n)\}$ represents the prior distribution $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$.

2. Upon the reception of observation \mathbf{z}_k at time k , a likelihood weighting is calculated for each prior sample $\mathbf{x}_k^-(i)$ based on the PDF $p(\mathbf{z}_k|\mathbf{x}_k^-(i))$ (of Equation D.6). As both \mathbf{z}_k and $\mathbf{x}_k^-(i)$ are specified, the value of $p(\mathbf{z}_k|\mathbf{x}_k^-(i))$ is scalar and $\Lambda_k(i)$ is given by

$$\Lambda_k(i) = \frac{p(\mathbf{z}_k|\mathbf{x}_k^-(i))}{\sum_{j=1}^n p(\mathbf{z}_k|\mathbf{x}_k^-(j))} \quad (\text{D.8})$$

That is, the likelihood $\Lambda_k(i)$ is proportional to $p(\mathbf{z}_k|\mathbf{x}_k^-(i))$ and normalised so that the sum of likelihoods is one.

3. Having assigned a weighting to each prior sample, the posterior $p(\mathbf{x}_k|\mathbf{Z}_k)$ is estimated by resampling from the prior sample set according to their weightings. The resampling algorithm is as follows. First, a set of cumulative likelihoods $\{\Upsilon_k(0), \dots, \Upsilon_k(n)\}$ is calculated.

$$\Upsilon_k(i) = \begin{cases} 0, & i = 0 \\ \sum_{j=1}^i \Lambda_k(j), & \text{otherwise} \end{cases} \quad (\text{D.9})$$

⁵Particle filters appear in the literature under a variety of guises including Bootstrap filter [60], SIR filter [114], Monte Carlo filter [79], and Condensation algorithm [74].

⁶The noise sequence \mathbf{q}_k is assumed to have known distribution $p(\mathbf{q}_k)$ at each time step k . It is also assumed that a random sample may be feasibly drawn from this distribution.

Each posterior sample $\mathbf{x}_k^+(i)$ is then found by drawing a random scalar sample u_i from the uniform distribution $\mathcal{U}(0, 1]$ and selecting the indicated sample from the the prior set.

$$\mathbf{x}_k^+(i) = \mathbf{x}_k^-(j), \quad \text{where } \Upsilon_k(j-1) < u_i \leq \Upsilon_k(j) \quad (\text{D.10})$$

The set of samples $\{\mathbf{x}_k^+(1), \dots, \mathbf{x}_k^+(n)\}$ then represents the posterior distribution $p(\mathbf{x}_k|\mathbf{Z}_k)$ and the estimation process can be repeated from step 1.

Implementation of step one is trivial. For each sample $\mathbf{x}_{k-1}(i)$ of the old posterior $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$, draw a sample $\mathbf{q}_{k-1}(i)$ from the distribution $p(\mathbf{q}_{k-1})$ and pass them through the state transition function. Steps two and three are less intuitive and the examples below elaborate on the calculation of the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k^-(i))$ for step two and implementation of the resampling phase for step three.

Example D.1

Likelihood weighting calculation. This example is taken from the first example in [60] and serves to elucidate the calculation of $p(\mathbf{z}_k|\mathbf{x}_k)$. Given a one-dimensional observation model

$$z_k = x_k^2/20 + r_k$$

where r_k is zero-mean Gaussian white noise with variance 1.0, the likelihood function $p(z_k|x_k)$ is found from Equation D.6 as follows.

$$p(z_k|x_k) = \int_{-\infty}^{\infty} \delta\left(z_k - \frac{x_k^2}{20} - r_k\right) p(r_k) dr_k$$

This represents, for specific values of $z_k = z_o$ and $x_k = x_o$, the area under the function resulting from the multiplication of $p(r_k)$ with the impulse $\delta(z_o - x_o^2/20 - r_k)$, where the impulse is non-zero at $r_k = z_o - x_o^2/20$ as shown in Figure D.1. In this way, given an observation z_o at time k , the likelihood of each sampled particle $\mathbf{x}_k^-(i)$ can be calculated.

The likelihood $p(z_k|x_k)$ can be expressed as a function of z_k in sensor-space for a fixed value of $x_k = x_o$ (see Figure D.2(a)). Alternatively, for a fixed value of $z_k = z_o$, the likelihood can be expressed as a function of x_k in target-space as shown in Figure D.2(b).

For general state-space systems with likelihood function defined by Equation D.6, the function $p(\mathbf{z}_k|\mathbf{x}_k = \mathbf{x}_o)$ always integrates to one meaning that the likelihood function is a PDF in sensor-space. However, in target space, the function $p(\mathbf{z}_k = \mathbf{z}_o|\mathbf{x}_k)$ does not in general integrate to one and so does not represent a PDF. For further information on likelihood functions, a detailed discussion can be found in [125, pages 33–35].

Example D.2

Resampling from the weighted prior sample set. Let \mathbf{x} be a one-dimensional random variable and let the prior PDF $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$ be represented by five samples

$$\{\mathbf{x}_k^-(1) = 6, \mathbf{x}_k^-(2) = 3, \mathbf{x}_k^-(3) = 8, \mathbf{x}_k^-(4) = 5, \mathbf{x}_k^-(5) = 7\}$$

with normalised likelihood weightings

$$\{\Lambda_k(1) = 0.35, \Lambda_k(2) = 0.05, \Lambda_k(3) = 0.15, \Lambda_k(4) = 0.2, \Lambda_k(5) = 0.25\}$$

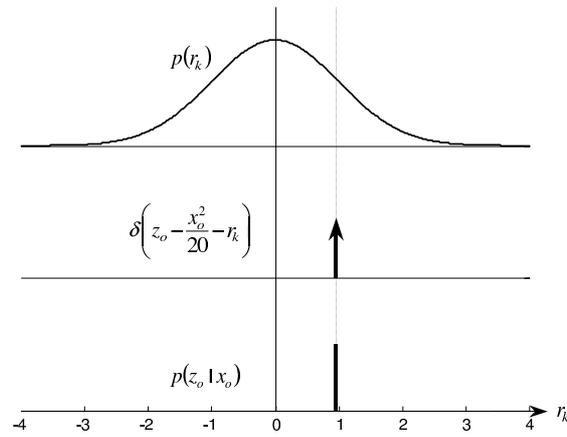


Figure D.1: Likelihood evaluation. For fixed values of z_k and x_k ($z_o = 5$ and $x_o = 9$ in this example), the likelihood is found as the area under the function obtained from multiplying $p(r_k)$ with the impulse function. Note, this area is simply the value of $p(r_k)$ at the location where $\delta(\cdot)$ is non-zero.

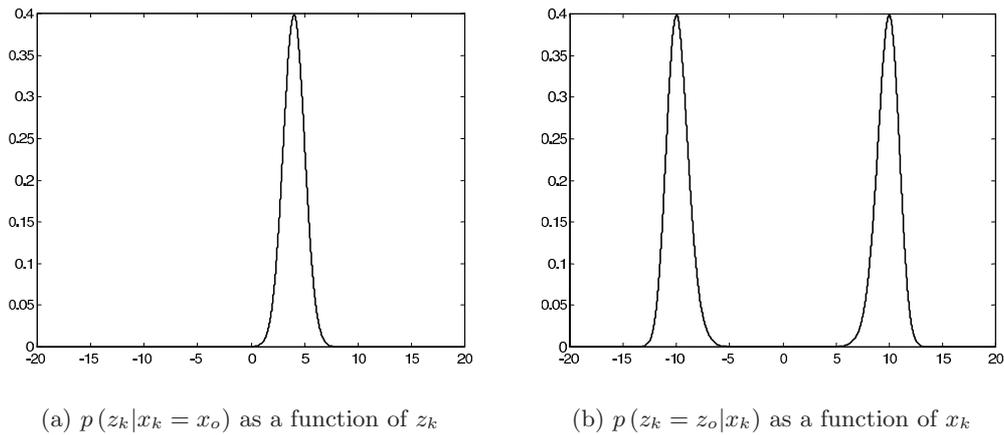


Figure D.2: Likelihood functions. The likelihood $p(\mathbf{z}_k | \mathbf{x}_k)$ for the model $z_k = x_k^2/20 + r_k$ is shown in (a) as a function of z_k with $x_o = 9$ and in (b) as a function of x_k with $z_o = 5$.

Algorithm D.1: RESAMPLE($\mathbf{x}_k^-, \Lambda_k$)

```

 $\Upsilon_1 \leftarrow \Lambda_1$ 
for  $i \leftarrow 2$  to  $n$ 
  do  $\Upsilon_i \leftarrow \Upsilon_{i-1} + \Lambda_i$ 
for  $i \leftarrow 1$  to  $n$ 
  do  $\Upsilon_i \leftarrow \Upsilon_i / \Upsilon_n$ 
 $U \leftarrow \text{UNIFORMRANDOM}(n)$ 
 $j \leftarrow 1$ 
for  $i \leftarrow 1$  to  $n$ 
  do  $\left\{ \begin{array}{l} \text{while } \Upsilon_j < u_i \\ \text{do } j \leftarrow j + 1 \\ \mathbf{x}_k^+(i) \leftarrow \mathbf{x}_k^-(j) \end{array} \right.$ 
return  $(\mathbf{x}_k^+)$ 

```

The cumulative likelihoods are therefore

$$\{\Upsilon_k(0) = 0, \Upsilon_k(1) = 0.35, \Upsilon_k(2) = 0.4, \Upsilon_k(3) = 0.55, \Upsilon_k(4) = 0.75, \Upsilon_k(5) = 1\}$$

A scalar value $u_1 = 0.43$ is drawn at random from a uniform distribution $\mathcal{U}(0, 1]$. This value is greater than $\Upsilon_k(2)$ and less than $\Upsilon_k(3)$ so the new sample $\mathbf{x}_k^+(1)$ is assigned the value of $\mathbf{x}_k^-(3)$. Suppose the five random scalars drawn were

$$\{u_1 = 0.43, u_2 = 0.78, u_3 = 0.60, u_4 = 0.29, u_5 = 0.85\}$$

then the posterior sample set would be as follows.

$$\{\mathbf{x}_k^+(1) = 8, \mathbf{x}_k^+(2) = 7, \mathbf{x}_k^+(3) = 5, \mathbf{x}_k^+(4) = 6, \mathbf{x}_k^+(5) = 7\}$$

D.4 Resampling Implementations

This section presents an implementation of the resampling algorithm. The key component to this algorithm is the generation of an ordered set of uniform-random numbers in the range $(0, 1]$, and so three alternative implementations of this module are presented.

The inputs to the resampling algorithm (see Algorithm D.1) are the prior sample set $\mathbf{x}_k^- = \{\mathbf{x}_k^-(1), \dots, \mathbf{x}_k^-(n)\}$ and their associated (non-normalised) likelihood weightings $\Lambda_k = \{\Lambda_k(1), \dots, \Lambda_k(n)\}$. The algorithm returns the posterior sample set $\mathbf{x}_k^+ = \{\mathbf{x}_k^+(1), \dots, \mathbf{x}_k^+(n)\}$.

The function UNIFORMRANDOM(n) receives the number of samples required and returns this number of uniform-random samples, in ascending order, from the range $(0, 1]$. The first version of this function, shown in Algorithm D.2, performs straight-forward calculation

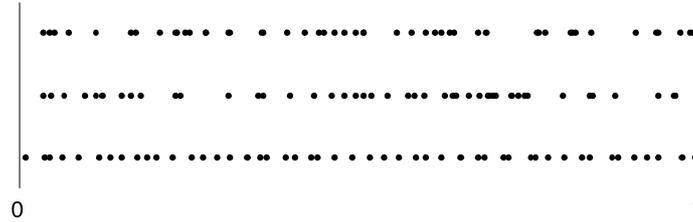


Figure D.3: Uniform-random sampling algorithms. The same set of 50 random samples is used in Algorithms D.2 to D.4 to allow comparison of their distributions (top to bottom, respectively). The first two algorithms possess identical statistical characteristics, while the third gives regularly spaced samples with uniform-random jitter in the range $(-\frac{1}{2n}, \frac{1}{2n})$.

Algorithm D.2: UNIFORMRANDOM(n)

```

for  $i \leftarrow 1$  to  $n$ 
  do  $u_i \leftarrow \mathcal{U}(0, 1]$ 
SORT( $U$ )
return ( $U$ )

```

Algorithm D.3: UNIFORMRANDOM(n)

```

 $u_n \leftarrow \mathcal{U}(0, 1]$ 
 $u_n \leftarrow u_n^{1/n}$ 
for  $i \leftarrow n - 1$  to  $1$ 
  do  $\begin{cases} u_i \leftarrow \mathcal{U}(0, 1] \\ u_i \leftarrow u_i^{1/i} u_{i+1} \end{cases}$ 
return ( $U$ )

```

Algorithm D.4: UNIFORMRANDOM(n)

```

 $k \leftarrow 1/n$ 
 $u_1 \leftarrow k/2$ 
for  $i \leftarrow 2$  to  $n$ 
  do  $\begin{cases} u_i \leftarrow u_{i-1} + k \\ u_{i-1} \leftarrow u_{i-1} + \mathcal{U}(-\frac{k}{2}, \frac{k}{2}) \end{cases}$ 
 $u_n \leftarrow u_n + \mathcal{U}(-\frac{k}{2}, \frac{k}{2})$ 
return ( $U$ )

```

of this set; first calculating a set of n values and later sorting them. The sort routine means that this implementation has $O(n \log n)$ complexity. A faster version with equivalent results is presented in Algorithm D.3, where the samples are created in order by way of exponential distributions—thus requiring only $O(n)$ time (this algorithm is also provided in [44, 111]). Finally, a method known as *stratified sampling* (see also [79]) is presented in Algorithm D.4. This algorithm also runs in $O(n)$ time, but tends to give better statistical results by ensuring that the samples are evenly distributed (i.e., without clustering), with spacing between samples not greater than $1/n$. A pictorial comparison of these algorithms is shown in Figure D.3.

D.5 Deficiencies of the Particle Filter Algorithm

Particle filter theory states that the sampled representation of posterior approaches the true posterior PDF as the number of samples approaches infinity. Practical implementations, however, are limited to a bounded number of samples and, in relation to this issue, the basic particle filter algorithm suffers from several important weaknesses.

The first of these is that the number of samples n required to obtain reasonable statistical results must typically be very large (even for state vectors of just one or two variables). Nevertheless, for many applications, the distribution statistics may not be crucial provided sample coverage is sufficient to ensure that the target state remains consistently trackable (i.e., mode tracking). But even this task becomes infeasible for larger state-spaces as the volume of state-space increases exponentially with the dimension of the state.

A second problem is sample impoverishment where the number of independent samples decreases over time (i.e., the filter behaves as if using only $n_e < n$ samples). This is apparent from Equation D.10 as clearly some samples from the prior set will be selected multiple times while others will be lost—resulting in a depletion in information. Sample impoverishment is countered in [60] by introducing “roughening”—adding Gaussian *jitter* to each resampled element $\mathbf{x}_k^+(i)$ —which seems to effectively eliminate sample dependence.

A third weakness arises when there is insufficient overlap between the prior sample set and the observation likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ so that the posterior estimate is very imprecise. This can happen, for example, if the likelihood function is highly peaked (as for a very accurate sensor), or if the area covered by the bulk of the likelihood function is on the outskirts of the prior sample set (as would occur for an outlier observation). The likelihood coverage problem is difficult to solve and requires directing resampling to regions where the observation likelihood is significant and weighting the results to maintain the correct posterior statistics. Methods for tackling this problem can be found in [60, 23, 52, 111, 138, 132].

Appendix E

Volume from Gaussian Multiplication

The sum of Gaussians cross-correlation operation in Chapter 4 requires evaluation of the volume¹ under the product of two Gaussians. A derivation of the volume equation is provided here in two sections. The first section shows the equation for the scaled Gaussian resulting from the multiplication of two Gaussians, and the second section solves for the volume under this Gaussian.

E.1 Multiplication of Two Gaussians

Let $g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})$ and $g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q})$ define two n -dimensional Gaussians

$$g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P}) = \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{P}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{P}^{-1}(\mathbf{x} - \bar{\mathbf{p}})\right) \quad (\text{E.1})$$

$$g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q}) = \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{Q}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{q}})^T \mathbf{Q}^{-1}(\mathbf{x} - \bar{\mathbf{q}})\right) \quad (\text{E.2})$$

where $\bar{\mathbf{p}}$, \mathbf{P} and $\bar{\mathbf{q}}$, \mathbf{Q} are their respective means and variances.² The scaled Gaussian $\alpha g(\mathbf{x}; \bar{\mathbf{r}}, \mathbf{R})$ resulting from the multiplication of these two Gaussians is therefore

$$\begin{aligned} \alpha g(\mathbf{x}; \bar{\mathbf{r}}, \mathbf{R}) &= g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q}) \\ &= \frac{\alpha}{(2\pi)^{n/2} \sqrt{|\mathbf{R}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{r}})^T \mathbf{R}^{-1}(\mathbf{x} - \bar{\mathbf{r}})\right) \end{aligned} \quad (\text{E.3})$$

where $\bar{\mathbf{r}}$ and \mathbf{R} are given by

$$\mathbf{R}^{-1} = \mathbf{P}^{-1} + \mathbf{Q}^{-1} \quad (\text{E.4})$$

$$\bar{\mathbf{r}} = \mathbf{R}(\mathbf{P}^{-1}\bar{\mathbf{p}} + \mathbf{Q}^{-1}\bar{\mathbf{q}}) \quad (\text{E.5})$$

and the scaling factor α represents the volume under the function $\alpha g(\mathbf{x}; \bar{\mathbf{r}}, \mathbf{R})$.

¹The term *volume* is used here to intuitively describe the integral over n -dimensional space.

²The means are $n \times 1$ real vectors and the variances are $n \times n$ real positive-definite matrices.

Proof. The derivation of Equation E.3 is as follows.³ From Equations E.1 and E.2, the multiplication of $g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})$ and $g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q})$ can be expanded to

$$g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q}) = \frac{1}{(2\pi)^n \sqrt{|\mathbf{P}||\mathbf{Q}|}} \exp\left(-\frac{1}{2} [(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{P}^{-1} (\mathbf{x} - \bar{\mathbf{p}}) + (\mathbf{x} - \bar{\mathbf{q}})^T \mathbf{Q}^{-1} (\mathbf{x} - \bar{\mathbf{q}})]\right) \quad (\text{E.6})$$

The next section of the proof concerns only the exponential portion of Equation E.6. For notational convenience, let $\mathbf{A} = \mathbf{P}^{-1}$ and $\mathbf{B} = \mathbf{Q}^{-1}$ so that

$$(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{P}^{-1} (\mathbf{x} - \bar{\mathbf{p}}) + (\mathbf{x} - \bar{\mathbf{q}})^T \mathbf{Q}^{-1} (\mathbf{x} - \bar{\mathbf{q}}) = (\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{A} (\mathbf{x} - \bar{\mathbf{p}}) + (\mathbf{x} - \bar{\mathbf{q}})^T \mathbf{B} (\mathbf{x} - \bar{\mathbf{q}}) \quad (\text{E.7})$$

Notice that $(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{A} (\mathbf{x} - \bar{\mathbf{p}})$ can be expanded using the matrix identity that $\mathbf{x}^T \mathbf{A} \mathbf{y} = \mathbf{y}^T \mathbf{A} \mathbf{x}$ if \mathbf{A} is real symmetric and $\mathbf{x}^T \mathbf{A} \mathbf{y}$ is scalar.

$$\begin{aligned} (\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{A} (\mathbf{x} - \bar{\mathbf{p}}) &= \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{A} \bar{\mathbf{p}} - \bar{\mathbf{p}}^T \mathbf{A} \mathbf{x} + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} \\ &= \mathbf{x}^T \mathbf{A} \mathbf{x} - 2\mathbf{x}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} \end{aligned}$$

Therefore, Equation E.7 becomes

$$\begin{aligned} &(\mathbf{x} - \bar{\mathbf{p}})^T \mathbf{A} (\mathbf{x} - \bar{\mathbf{p}}) + (\mathbf{x} - \bar{\mathbf{q}})^T \mathbf{B} (\mathbf{x} - \bar{\mathbf{q}}) \\ &= \mathbf{x}^T \mathbf{A} \mathbf{x} - 2\mathbf{x}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} + \mathbf{x}^T \mathbf{B} \mathbf{x} - 2\mathbf{x}^T \mathbf{B} \bar{\mathbf{q}} + \bar{\mathbf{q}}^T \mathbf{B} \bar{\mathbf{q}} \\ &= \mathbf{x}^T (\mathbf{A} + \mathbf{B}) \mathbf{x} - 2\mathbf{x}^T (\mathbf{A} \bar{\mathbf{p}} + \mathbf{B} \bar{\mathbf{q}}) + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T \mathbf{B} \bar{\mathbf{q}} \end{aligned}$$

(substituting $\mathbf{C} = \mathbf{R}^{-1}$ from Equation E.4)

$$= \mathbf{x}^T \mathbf{C} \mathbf{x} - 2\mathbf{x}^T (\mathbf{A} \bar{\mathbf{p}} + \mathbf{B} \bar{\mathbf{q}}) + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T \mathbf{B} \bar{\mathbf{q}}$$

(inserting $\mathbf{C} \mathbf{C}^{-1}$)

$$= \mathbf{x}^T \mathbf{C} \mathbf{x} - 2\mathbf{x}^T \mathbf{C} \mathbf{C}^{-1} (\mathbf{A} \bar{\mathbf{p}} + \mathbf{B} \bar{\mathbf{q}}) + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T \mathbf{B} \bar{\mathbf{q}}$$

(substituting $\bar{\mathbf{r}}$ from Equation E.5)

$$= \mathbf{x}^T \mathbf{C} \mathbf{x} - 2\mathbf{x}^T \mathbf{C} \bar{\mathbf{r}} + \bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T \mathbf{B} \bar{\mathbf{q}}$$

(inserting $\bar{\mathbf{r}}^T \mathbf{C} \bar{\mathbf{r}} - \bar{\mathbf{r}}^T \mathbf{C} \bar{\mathbf{r}}$)

$$\begin{aligned} &= \mathbf{x}^T \mathbf{C} \mathbf{x} - 2\mathbf{x}^T \mathbf{C} \bar{\mathbf{r}} + \bar{\mathbf{r}}^T \mathbf{C} \bar{\mathbf{r}} + \underbrace{\bar{\mathbf{p}}^T \mathbf{A} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T \mathbf{B} \bar{\mathbf{q}} - \bar{\mathbf{r}}^T \mathbf{C} \bar{\mathbf{r}}}_k \\ &= (\mathbf{x} - \bar{\mathbf{r}})^T \mathbf{C} (\mathbf{x} - \bar{\mathbf{r}}) + k \end{aligned}$$

³Thanks to Somajyoti Majumder for his assistance in this derivation. Note, the standard proof for “the product of two Gaussians is a Gaussian” is rather simpler than this one, but this approach fulfills a secondary motivation to obtain Equation E.9.

where k is a scalar constant term. Thus, incorporating this result back into Equation E.6 the multiplication of $g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})$ and $g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q})$ can be expressed as

$$\begin{aligned} g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q}) &= \frac{1}{(2\pi)^n \sqrt{|\mathbf{P}||\mathbf{Q}|}} \exp\left(-\frac{1}{2} [(\mathbf{x} - \bar{\mathbf{r}})^T \mathbf{R}^{-1} (\mathbf{x} - \bar{\mathbf{r}}) + k]\right) \\ &= \frac{1}{(2\pi)^n \sqrt{|\mathbf{P}||\mathbf{Q}|}} \exp\left(-\frac{k}{2}\right) \exp\left(-\frac{1}{2} (\mathbf{x} - \bar{\mathbf{r}})^T \mathbf{R}^{-1} (\mathbf{x} - \bar{\mathbf{r}})\right) \end{aligned} \quad (\text{E.8})$$

which is equivalent to Equation E.3 for α equal to

$$\alpha = \frac{(2\pi)^{n/2} \sqrt{|\mathbf{R}|}}{(2\pi)^n \sqrt{|\mathbf{P}||\mathbf{Q}|}} \exp\left(-\frac{k}{2}\right) \quad (\text{E.9})$$

E.2 Volume from the Multiplication of Two Gaussians

The volume α under the scaled Gaussian resulting from the multiplication of $g(\mathbf{x}; \bar{\mathbf{p}}, \mathbf{P})$ and $g(\mathbf{x}; \bar{\mathbf{q}}, \mathbf{Q})$ is given in Equation E.9 above. This equation expressed in terms of $\bar{\mathbf{p}}, \mathbf{P}, \bar{\mathbf{q}}$ and \mathbf{Q} is as follows.

$$\alpha = \frac{1}{(2\pi)^{n/2} \sqrt{|\mathbf{P} + \mathbf{Q}|}} \exp\left(-\frac{1}{2} (\bar{\mathbf{p}} - \bar{\mathbf{q}})^T (\mathbf{P} + \mathbf{Q})^{-1} (\bar{\mathbf{p}} - \bar{\mathbf{q}})\right) \quad (\text{E.10})$$

Proof. The derivation of Equation E.10 from Equation E.9 is demonstrated in two parts. First, the simplification of

$$\begin{aligned} \frac{\sqrt{|\mathbf{R}|}}{\sqrt{|\mathbf{P}||\mathbf{Q}|}} &= \frac{\sqrt{|\mathbf{P}(\mathbf{P} + \mathbf{Q})^{-1}\mathbf{Q}|}}{\sqrt{|\mathbf{P}||\mathbf{Q}|}} \\ &= \sqrt{|\mathbf{P} + \mathbf{Q}|^{-1}} \\ &= \frac{1}{\sqrt{|\mathbf{P} + \mathbf{Q}|}} \end{aligned} \quad (\text{E.11})$$

and, second, the simplification of the constant k .

$$k = \bar{\mathbf{p}}^T \mathbf{P}^{-1} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T \mathbf{Q}^{-1} \bar{\mathbf{q}} - \bar{\mathbf{r}}^T \mathbf{R}^{-1} \bar{\mathbf{r}} \quad (\text{E.12})$$

The portion $\bar{\mathbf{r}}^T \mathbf{R}^{-1} \bar{\mathbf{r}}$ is expanded as

$$\begin{aligned} \bar{\mathbf{r}}^T \mathbf{R}^{-1} \bar{\mathbf{r}} &= (\mathbf{P}^{-1} \bar{\mathbf{p}} + \mathbf{Q}^{-1} \bar{\mathbf{q}})^T \mathbf{R}^T \mathbf{R}^{-1} \mathbf{R} (\mathbf{P}^{-1} \bar{\mathbf{p}} + \mathbf{Q}^{-1} \bar{\mathbf{q}}) \\ &= (\bar{\mathbf{p}}^T \mathbf{P}^{-1} + \bar{\mathbf{q}}^T \mathbf{Q}^{-1}) (\mathbf{P}(\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q}) (\mathbf{P}^{-1} \bar{\mathbf{p}} + \mathbf{Q}^{-1} \bar{\mathbf{q}}) \\ &= \bar{\mathbf{p}}^T \mathbf{P}^{-1} \mathbf{P} (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q} \mathbf{P}^{-1} \bar{\mathbf{p}} \\ &\quad + \bar{\mathbf{q}}^T \mathbf{Q}^{-1} \mathbf{Q} (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{P} \mathbf{Q}^{-1} \bar{\mathbf{q}} \\ &\quad + \bar{\mathbf{p}}^T \mathbf{P}^{-1} \mathbf{P} (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q} \mathbf{Q}^{-1} \bar{\mathbf{q}} \\ &\quad + \bar{\mathbf{q}}^T \mathbf{Q}^{-1} \mathbf{Q} (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{P} \mathbf{P}^{-1} \bar{\mathbf{p}} \\ &= \bar{\mathbf{p}}^T (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q} \mathbf{P}^{-1} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{P} \mathbf{Q}^{-1} \bar{\mathbf{q}} \\ &\quad + \bar{\mathbf{p}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{q}} + \bar{\mathbf{q}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{p}} \end{aligned}$$

so that Equation E.12 becomes

$$\begin{aligned} k &= \bar{\mathbf{p}}^T (\mathbf{P}^{-1} - (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q} \mathbf{P}^{-1}) \bar{\mathbf{p}} \\ &\quad + \bar{\mathbf{q}}^T (\mathbf{Q}^{-1} - (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{P} \mathbf{Q}^{-1}) \bar{\mathbf{q}} \\ &\quad - \bar{\mathbf{p}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{q}} - \bar{\mathbf{q}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{p}} \end{aligned}$$

Now, $\mathbf{P}^{-1} - (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q} \mathbf{P}^{-1}$ can be simplified as follows.

$$\begin{aligned} \mathbf{P}^{-1} - (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{Q} \mathbf{P}^{-1} &= (\mathbf{P} + \mathbf{Q})^{-1} ((\mathbf{P} + \mathbf{Q}) \mathbf{P}^{-1} - \mathbf{Q} \mathbf{P}^{-1}) \\ &= (\mathbf{P} + \mathbf{Q})^{-1} (\mathbf{P} \mathbf{P}^{-1} + \mathbf{Q} \mathbf{P}^{-1} - \mathbf{Q} \mathbf{P}^{-1}) \\ &= (\mathbf{P} + \mathbf{Q})^{-1} \end{aligned}$$

Similarly, $\mathbf{Q}^{-1} - (\mathbf{P} + \mathbf{Q})^{-1} \mathbf{P} \mathbf{Q}^{-1} = (\mathbf{P} + \mathbf{Q})^{-1}$ so that Equation E.12 becomes

$$\begin{aligned} k &= \bar{\mathbf{p}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{p}} + \bar{\mathbf{q}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{q}} \\ &\quad - \bar{\mathbf{p}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{q}} - \bar{\mathbf{q}}^T (\mathbf{P} + \mathbf{Q})^{-1} \bar{\mathbf{p}} \\ &= (\bar{\mathbf{p}} - \bar{\mathbf{q}})^T (\mathbf{P} + \mathbf{Q})^{-1} (\bar{\mathbf{p}} - \bar{\mathbf{q}}) \end{aligned} \tag{E.13}$$

Substituting the results of Equations E.11 and E.13 into Equation E.9 provides the solution given in Equation E.10.

Bibliography

- [1] S. Argamon-Engelson. Using image signatures for place recognition. *Pattern Recognition Letters*, 19:941–951, 1998.
- [2] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.
- [3] O. Aycard, F. Charpillet, D. Fohr, and J.F. Mari. Place learning and recognition using hidden markov models. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1741–1746, 1997.
- [4] T. Bailey and E. Nebot. Localisation in large-scale environments. *Robotics and Autonomous Systems*, 37(4):261–281, 2001.
- [5] T. Bailey, E.M. Nebot, J.K. Rosenblatt, and H.F. Durrant-Whyte. Data association for mobile robot navigation: A graph theoretic approach. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2512–2517, 2000.
- [6] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.
- [7] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, 1975.
- [8] H.G. Barrow and R.M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4(4):83–84, 1976.
- [9] K. Basye, T. Dean, and J.S. Vitter. Coping with uncertainty in map learning. Technical report, Brown University, Department of Computer Science, 1989.
- [10] R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. *Algorithmica*, 29(4):610–637, 2001.
- [11] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.
- [12] J.L. Bentley. K-d trees for semidynamic point sets. In *Sixth Annual ACM Symposium on Computational Geometry*, pages 187–197, 1990.
- [13] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

-
- [14] S.S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House Radar Library, 1999.
- [15] M. Boddy and T. Dean. Solving time-dependent planning problems. In *Eleventh International Joint Conference on Artificial Intelligence*, pages 979–984, 1989.
- [16] R.C. Bolles and R.A. Cain. Recognizing and locating partially visible objects: The local-feature-focus method. *International Journal of Robotics Research*, 1(3):57–82, 1982.
- [17] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo. The maximum clique problem. In D.Z. Du and P.M. Pardalos, editors, *Handbook of Combinatorial Optimization*. Kluwer Academic Publishers, 1999.
- [18] M. Bosse, P. Newman, J. Leonard, and S. Teller. An Atlas framework for scalable mapping. Technical report, Massachusetts Institute of Technology, Marine Robotics Laboratory, 2002.
- [19] R.N. Bracewell. *The Fourier Transform and its Applications, 2nd Edition*. McGraw-Hill, 1986.
- [20] M. Brady, S. Cameron, H. Durrant-Whyte, M. Fleck, D. Forsyth, A. Noble, and I. Page. Progress towards a system that can acquire pallets and clean warehouses. In *Fourth International Symposium of Robotics Research*, pages 359–374, 1987.
- [21] W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.
- [22] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping with mobile robots using EM. In *International Conference on Machine Learning*, 1999.
- [23] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. Technical report, Oxford University, Department of Statistics, 1997.
- [24] J.A. Castellanos, M. Devy, and J.D. Tardós. Towards a topological representation of indoor environments: A landmark-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 23–28, 1999.
- [25] J.A. Castellanos, J.M.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*, 15(5):948–952, 1999.
- [26] J.A. Castellanos and J.D. Tardós. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, 1999.
- [27] J.A. Castellanos, J.D. Tardós, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *IEEE International Conference on Robotics and Automation*, pages 1053–1059, 1997.

-
- [28] C.K. Chen and D.Y.Y. Yun. Unify graph matching problems with a practical solution. In *International Conference on Systems, Signals, Control, Computers*, 1998.
- [29] K.S. Chong and L. Kleeman. Large scale sonarray mapping using multiple connected local maps. In *International Conference on Field and Service Robotics*, pages 538–545, 1997.
- [30] K.S. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal Robotics Research*, 18(1):3–19, 1999.
- [31] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, 2001.
- [32] J.B. Collins and J.K. Uhlmann. Efficient gating in data association with multivariate gaussian distributed states. *IEEE Transactions on Aerospace and Electronic Systems*, 28(3):909–916, 1992.
- [33] I.J. Cox. Blanche—An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, pages 193–204, 1991.
- [34] I.J. Cox and S.L. Hingorani. An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.
- [35] I.J. Cox and M.L. Miller. On finding ranked assignments with application to multi-target tracking and motion correspondence. *IEEE Transactions on Aerospace and Electronic Systems*, pages 486–489, 1995.
- [36] M. Csorba. *Simultaneous Localisation and Map Building*. PhD thesis, University of Oxford, Department of Engineering Science, 1997.
- [37] J. de Geeter, H. van Brussel, J. de Schutter, and M. Decréton. A smoothly constrained Kalman filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1171–1177, 1997.
- [38] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 1322–1328, 1999.
- [39] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [40] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1009–1014, 2000.
- [41] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte. A new algorithm for the alignment of inertial measurement units without external observation for land vehicle applications. In *IEEE International Conference on Robotics and Automation*, pages 2274–2279, 1999.

-
- [42] M.W.M.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*, 17(3):229–241, 2001.
- [43] C. Dorai, J. Weng, and A.K. Jain. Optimal registration of object views using range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1131–1138, 1997.
- [44] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical report, Cambridge University, Department of Engineering, 1998.
- [45] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [46] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Map validation and robot self-location in a graph-like world. *Robotics and Autonomous Systems*, 22(2):159–178, 1997.
- [47] A. Dunkin. Automated guided vehicle systems: An introduction. *Industrial Engineering*, 26:47–53, August 1994.
- [48] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheduling. A Bayesian algorithm for simultaneous localisation and map building. In *Tenth International Symposium of Robotics Research*, 2001.
- [49] H.F. Durrant-Whyte. An autonomous guided vehicle for cargo handling applications. *International Journal of Robotics Research*, 15, 1996.
- [50] A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception. In *Sixth Conference on Uncertainty in AI*, 1990.
- [51] S.P. Engelson and D.V. McDermott. Error correction in mobile robot map learning. In *IEEE International Conference on Robotics and Automation*, pages 2555–2560, 1992.
- [52] P. Fearnhead. *Sequential Monte Carlo Methods in Filter Theory*. PhD thesis, University of Oxford, 1998.
- [53] H.J.S. Feder, J.J. Leonard, and C.M. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7):650–668, 1999.
- [54] T.A. Feo, M.G.C. Resende, and S.H. Smith. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42:860–878, 1994.
- [55] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983.
- [56] Y. Fuke and E. Krotkov. Dead reckoning for a lunar rover on uneven terrain. In *IEEE International Conference on Robotics and Automation*, pages 411–416, 1996.

-
- [57] G. Giralt, R. Sobek, and R. Chatila. A multi-level planning and navigation system for a mobile robot: A first approach to Hilare. In *Sixth International Joint Conference on Robotics and Automation*, 1979.
- [58] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [59] A.V. Goldberg and R. Kennedy. An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 71:153–178, 1995.
- [60] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.
- [61] M.S. Grewal and A.P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB, 2nd Edition*. John Wiley and Sons, 2001.
- [62] M.S. Grewal, L.R. Weill, and A.P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons, 2001.
- [63] W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.
- [64] J. Guivant and E. Nebot. Navigation in large outdoor unstructured environments. In *Tenth International Symposium of Robotics Research*, 2001.
- [65] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.
- [66] J. Guivant and E. Nebot. Improving computational and memory requirements of simultaneous localization and map building algorithms. In *IEEE International Conference on Robotics and Automation*, pages 2731–2736, 2002.
- [67] J. Guivant, E. Nebot, and H.F. Durrant-Whyte. Simultaneous localization and map building using natural features in outdoor environments. In *Sixth International Conference on Intelligent Autonomous Systems*, volume 1, pages 581–588, 2000.
- [68] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, 1999.
- [69] J.S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *1st Euromicro Workshop on Advanced Mobile Robots (Eurobot'96)*, pages 61–67, 1996.
- [70] E. Haines. Point in polygon strategies. In P.S. Heckbert, editor, *Graphics Gems IV*, pages 24–46. Academic Press, 1994.
- [71] R. Hinkel and T. Knieriemen. Environment perception with a laser radar in a fast moving robot. In *Symposium on Robot Control*, pages 68.1–68.7, 1988.

-
- [72] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, 1989.
- [73] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [74] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *European Conference on Computer Vision*, pages 343–356, 1996.
- [75] 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 Automatic Control*, 45(3):477–482, 2000.
- [76] S.J. Julier and J.K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, University of Oxford, Department of Engineering Science, 1996.
- [77] S.J. Julier and J.K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference*, volume 4, pages 2369–2373, 1997.
- [78] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
- [79] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.
- [80] L. Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead reckoning. In *IEEE International Conference on Robotics and Automation*, pages 2582–2587, 1992.
- [81] J. Knight, A. Davison, and I. Reid. Towards constant time SLAM using postponement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 405–413, 2001.
- [82] K. Konolige and K. Chou. Markov localization using correlation. In *International Joint Conference on Artificial Intelligence*, pages 1154–1159, 1999.
- [83] D. Kortenkamp, L.D. Baker, and T. Weymouth. Using gateways to build a route map. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2209–2214, 1992.
- [84] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
- [85] B. Kuipers and Y.T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.

-
- [86] A.T. Le. *Modelling and Control of Tracked Vehicles*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 1999.
- [87] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE International Workshop on Intelligent Robots and Systems*, pages 1442–1447, 1991.
- [88] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992.
- [89] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous robot. *International Journal of Robotics Research*, 11(4):286–298, 1992.
- [90] J.J. Leonard and H.J.S. Feder. Decoupled stochastic mapping. Technical report, Massachusetts Institute of Technology, Marine Robotics Laboratory, 1999.
- [91] J.J. Leonard and R.J. Rikoski. Incorporation of delayed decision making into stochastic mapping. In *International Symposium On Experimental Robotics*, 2000.
- [92] F. Lu and E. Milius. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
- [93] F. Lu and E. Milius. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1997.
- [94] R. Madhavan. *Terrain Aided Localisation of Autonomous Vehicles in Unstructured Environments*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2001.
- [95] S. Majumder. *Sensor Fusion and Feature Based Navigation for Subsea Robots*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2001.
- [96] S. Majumder, J. Rosenblatt, S. Scheduling, and H. Durrant-Whyte. Map building and localization for underwater navigation. In *International Symposium On Experimental Robotics*, 2000.
- [97] P.S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, 1979.
- [98] J.J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *Software Practice and Experience*, 12:23–34, 1982.
- [99] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.
- [100] H.P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation*, pages 116–121, 1985.
- [101] D.M. Mount. ANN programming manual. Technical report, University of Maryland, Department of Computer Science, 1998.

-
- [102] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environmental modelling. In *Fifth International Symposium of Robotics Research*, pages 85–94, 1989.
- [103] E.M. Nebot and D. Pagac. Quadtree representation and ultrasonic information for mapping an autonomous guided vehicle’s environment. *International Journal of Computers and Their Applications*, 2(3):160–170, 1995.
- [104] J. Neira, L. Montano, and J.D. Tardós. Constraint-based object recognition in multisensor systems. In *IEEE International Conference on Robotics and Automation*, pages 135–142, 1993.
- [105] J. Neira and J.D. Tardós. Robust and feasible data association for simultaneous localization and map building. In *IEEE International Conference on Robotics and Automation, Workshop on Robot Navigation and Mapping*, 2000.
- [106] J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
- [107] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
- [108] C.F. Olson and L.H. Matthies. Maximum likelihood rover localization by matching range maps. In *IEEE International Conference on Robotics and Automation*, pages 272–277, 1998.
- [109] A. Papoulis. *Probability, Random Variables, and Stochastic Processes, 2nd Edition*. McGraw-Hill, 1984.
- [110] S.T. Pfister, K.L. Kriechbaum, S.I. Roumeliotis, and J.W. Burdick. Weighted range sensor matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation*, 2002.
- [111] M.K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94:590–599, 1999.
- [112] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C, 2nd Edition*. Cambridge University Press, 1992.
- [113] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
- [114] D.B. Rubin. Using the SIR algorithm to simulate posterior distributions. In J.M. Bernardo, M.H. DeGroot, D.V. Lindley, and A.F.M. Smith, editors, *Bayesian Statistics 3*, pages 395–402. Oxford University Press, 1988.
- [115] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling*, pages 145–152, 2001.

-
- [116] S. Scheduling, G. Dissanayake, E. Nebot, and H. Durrant-Whyte. Slip modelling and aided inertial navigation of an LHD. In *IEEE International Conference on Robotics and Automation*, pages 1904–1909, 1997.
- [117] S. Scheduling, E. Nebot, and H. Durrant-Whyte. High-integrity navigation: A frequency-domain approach. *IEEE Transactions on Control Systems Technology*, 8(4):676–694, 2000.
- [118] A.C. Schultz and W. Adams. Continuous localization using evidence grids. In *IEEE International Conference on Robotics and Automation*, pages 2833–2839, 1998.
- [119] A.C. Schultz, W. Adams, B. Yamauchi, and M. Jones. Unifying exploration, localization, navigation and planning through a common representation. In *IEEE International Conference on Robotics and Automation*, pages 2651–2658, 1999.
- [120] A. Shoukry and M. Aboutabl. Neural network approach for solving the maximal common subgraph problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(5):785–790, 1996.
- [121] S. Simhon and G. Dudek. A global topological map formed by local metric maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1708–1714, 1998.
- [122] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Fourth International Symposium of Robotics Research*, pages 467–474, 1987.
- [123] A. Stevens, M. Stevens, and H.F. Durrant-Whyte. OxNav: Reliable autonomous navigation. In *IEEE International Conference on Robotics and Automation*, pages 2607–2612, 1995.
- [124] A.J. Stoddart, S. Lemke, A. Hilton, and T. Renn. Estimating pose uncertainty for surface registration. *Image and Vision Computing*, 16(2):111–120, 1998.
- [125] L.D. Stone, C.A. Barlow, and T.L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999.
- [126] S. Sukkarieh, E.M. Nebot, and H.F. Durrant-Whyte. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3):572–578, 1999.
- [127] J.D. Tardós, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4):311–330, 2002.
- [128] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [129] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

-
- [130] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *International Conference on Robotics and Automation*, pages 321–328, 2000.
- [131] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998.
- [132] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128:99–141, 2001.
- [133] T. Tsumura. Survey of automated guided vehicle in japanese factory. In *IEEE International Conference on Robotics and Automation*, pages 1329–1334, 1986.
- [134] J.K. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, Department of Engineering Science, 1995.
- [135] J.K. Uhlmann, S.J. Julier, and M. Csorba. Nondivergent simultaneous map building and localization using covariance intersection. In *SPIE Aerosense Conference on Navigation and Control Technologies for Unmanned Systems II*, volume 3087, pages 2–11, 1997.
- [136] J.R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23(1):31–42, 1976.
- [137] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE International Conference on Robotics and Automation*, pages 1023–1029, 2000.
- [138] R. van der Merwe, J.F.G. de Freitas, D. Doucet, and E.A. Wan. The unscented particle filter. Technical report, Cambridge University, Department of Engineering, 2000.
- [139] G. Weiß, C. Wetzler, and E. Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *International Conference on Intelligent Robots and Systems*, pages 595–601, 1994.
- [140] S.B. Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2001.
- [141] B. Yamauchi and P. Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, 14(2):107–120, 1997.
- [142] B. Yamauchi, A. Schultz, and W. Adams. Mobile robot exploration and map-building with continuous localization. In *IEEE International Conference on Robotics and Automation*, pages 3715–3720, 1998.
- [143] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.