

# SIMULTANEOUS LOCALIZATION, MAPPING AND MOVING OBJECT TRACKING

**Chieh-Chih Wang**

CMU-RI-TR-04-23

Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

April 2004

*Submitted in partial fulfilment of  
the requirements for the degree of  
Doctor of Philosophy*

Thesis Committee:  
Charles Thorpe, Chair  
Martial Hebert  
Sebastian Thrun, Stanford University  
Hugh Durrant-Whyte, University of Sydney



## ABSTRACT

---

LOCALIZATION, mapping and moving object tracking serve as the basis for scene understanding, which is a key prerequisite for making a robot truly autonomous. Simultaneous localization, mapping and moving object tracking (SLAMMOT) involves not only simultaneous localization and mapping (SLAM) in dynamic environments but also detecting and tracking these dynamic objects. It is believed by many that a solution to the SLAM problem would open up a vast range of potential applications for autonomous robots. Accordingly, a solution to the SLAMMOT problem would expand robotic applications in proximity to human beings where robots work not only for people but also with people.

This thesis establishes a new discipline at the intersection of SLAM and moving object tracking. Its contributions are two-fold: theoretical and practical.

From a theoretical perspective, we establish a mathematical framework to integrate SLAM and moving object tracking, which provides a solid basis for understanding and solving the whole problem. We describe two solutions: SLAM with generic objects (GO), and SLAM with detection and tracking of moving objects (DATMO). SLAM with GO calculates a joint posterior over all generic objects and the robot. Such an approach is similar to existing SLAM algorithms, but with additional structure to allow for motion modelling of the generic objects. Unfortunately, it is computationally demanding and infeasible. Consequently, we provide the second solution, SLAM with DATMO, in which the estimation problem is decomposed into two separate estimators. By maintaining separate posteriors for the stationary objects and the moving objects, the resulting estimation problems are much lower dimensional than SLAM with GO.

From a practical perspective, we develop algorithms for dealing with the implementation issues on perception modelling, motion modelling and data association. Regarding perception modelling, a hierarchical object based representation is presented to integrate existing feature-based, grid-based and direct methods. The sampling- and correlation-based range image matching algorithm is developed to tackle the problems arising from uncertain, sparse and featureless measurements. With regard to motion modelling, we describe a move-stop hypothesis tracking algorithm to tackle the difficulties of tracking ground moving objects. Kinematic information from motion modelling as well as geometric information from perception modelling is used to aid data association at different levels. By following the theoretical guidelines and implementing the described algorithms, we are able to demonstrate the feasibility of SLAMMOT using data collected from the Navlab8 and Navlab11 vehicles at high speeds in crowded urban environments.



## ACKNOWLEDGEMENTS

---

First and foremost, I would like to thank my advisor Chuck Thorpe for supporting me throughout the years, and for his priceless technical advice and wisdom. Martial Hebert has long been an inspiration to me. Thanks to many enjoyable discussions with Martial during my Ph.D career. My gratitude also goes to Sebastian Thrun and Hugh Durrant-Whyte, whose suggestions, insights and critique proved to be invaluable to this thesis.

I would like to thank the members of the Navlab group for their excellent work on building and maintaining the Navlab8 and Navlab11 vehicles, and for their helps on collecting data. I would like to specifically acknowledge: Justin Carlson, David Duggins, Arne Suppe, John Kozar, Jay Gowdy, Robert MacLachlan, Christoph Mertz and David Duke.

Thanks also go to the member of the 3D computer vision group and the MISC reading group. The weekly meetings with the MISC reading group have proved to be one of my best learning experiences at CMU. I would like to specifically acknowledge Daniel Huber, Nicolas Vandapel, and Owen Carmichael for their helps on Spin Images.

I would like to thank my many friends at CMU with whom I have the pleasure of working and living over the years. These include Carl Wellington, Vandt Verma, Wei-Tech Ang, Cristian Dima, Wen-Chieh "Steve" Lin, Jing Xiao, Fernando Alfaro, Curt Bererton, Anthony Gallagher, Jinxiang Chai, Kiran Bhat, Aaron Courville, Siddhartha Srinivasa, Liang Zhao (now at Univ. of Maryland), Matt Deans (now at NASA Ames), and Stella Yu (now at Berkeley).

Thanks to the members of my research qualifier committee, John Bares, Simon Baker and Peng Chang (now at Sarnoff) for their feedbacks on earlier research.

I would like to thank Peter Cheeseman for hiring me as an intern at NASA Ames research center in 2002 to work on 3D SLAM, and Dirk Langer for hiring me as an intern in 2001 to work on Spin Images.

Thanks to Suzanne Lyons Muth for her greatly administrative support.

Special thanks go to my parents, my brother Chieh-Kun, and sister-in-law Tsu-Ying for their support and sacrifices, and for letting me pursue my dreams over the years.

Finally, I would like to thank my wife Jessica Hsiao-Ping for the weekly commutes between Columbus and Pittsburgh under all weathers, for providing a welcome distraction from school, for bringing me happiness, and for her support, encouragement and love.

This thesis was funded in part by the U.S. Department of Transportation; the Federal Transit Administration; by Bosch Corporation; and by SAIC Inc. Their support is gratefully acknowledged.



# TABLE OF CONTENTS

---

ABSTRACT . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	v
LIST OF FIGURES . . . . .	xi
LIST OF TABLES . . . . .	xvii
CHAPTER 1. Introduction . . . . .	1
1.1. Safe Driving . . . . .	3
Localization . . . . .	3
Simultaneous Localization and Mapping . . . . .	3
Detection and Tracking of Moving Objects . . . . .	3
SLAM vs. DATMO . . . . .	4
1.2. City-Sized Simultaneous Localization and Mapping . . . . .	5
Computational Complexity . . . . .	6
Representation . . . . .	7
Data Association in the Large . . . . .	7
1.3. Moving Object Tracking in Crowded Urban Environments . . . . .	8
Detection . . . . .	8
Cluttered Environments . . . . .	8
Motion Modelling . . . . .	8
1.4. Simultaneous Localization, Mapping and Moving Object Tracking . . . . .	9
1.5. Experimental Setup . . . . .	9
1.6. Thesis Statement . . . . .	10
1.7. Document Outline . . . . .	12
CHAPTER 2. Foundations . . . . .	15
2.1. Uncertain Spatial Relationships . . . . .	17
Compounding . . . . .	17
The Inverse Relationship . . . . .	18
The Tail-to-Tail Relationship . . . . .	19
Unscented Transform . . . . .	19
2.2. Simultaneous Localization and Mapping . . . . .	19
Formulation of SLAM . . . . .	20
Calculation Procedures . . . . .	20
Computational Complexity . . . . .	25
Perception Modelling and Data Association . . . . .	26
2.3. Moving Object Tracking . . . . .	26
Formulation of Moving Object Tracking . . . . .	27
Mode Learning and State Inference . . . . .	27

TABLE OF CONTENTS

Calculation Procedures of the IMM algorithm . . . . .	33
Motion Modelling . . . . .	35
Perception Modelling and Data Association . . . . .	36
2.4. SLAM with Generic Objects . . . . .	36
2.5. SLAM with Detection and Tracking of Moving Objects . . . . .	38
Formulation of SLAM with DATMO . . . . .	38
Calculation Procedures . . . . .	41
2.6. Summary . . . . .	44
CHAPTER 3. Perception Modelling . . . . .	45
3.1. Perception Models . . . . .	46
Feature-based methods . . . . .	46
Grid-based methods . . . . .	46
Direct methods . . . . .	48
Comparison . . . . .	49
3.2. Hierarchical Object based Representation . . . . .	49
Scan Segmentation . . . . .	51
Perception Sensor Modelling . . . . .	52
Sparse Data . . . . .	53
3.3. Sampling- and Correlation-based Range Image Matching . . . . .	54
The Iterated Closest Point Algorithm . . . . .	54
Correspondence Finding Ambiguity . . . . .	57
Measurement Noises . . . . .	59
Object Saliency Score . . . . .	61
3.4. Hierarchical Object-based Representation for Tracking . . . . .	62
3.5. Hierarchical Object-based SLAM . . . . .	64
Local Mapping using Grid-based approaches . . . . .	65
Global Mapping using Feature-based Approaches . . . . .	67
3.6. Summary . . . . .	70
CHAPTER 4. Motion Modelling . . . . .	73
4.1. Model Selection . . . . .	73
Off-line Learning . . . . .	74
Online Adapting . . . . .	74
4.2. Robot Motion Modelling . . . . .	75
4.3. Moving Object Motion Modelling . . . . .	75
Discrete Time State Space Model . . . . .	76
The Constant Velocity Model . . . . .	76
The Constant Acceleration Model . . . . .	77
The IMM algorithm with the CV and CA Models . . . . .	78
4.4. The Stationary Motion Model . . . . .	81
The Stop Model Simplified from the CV Model . . . . .	81
The Stationary Process Model . . . . .	82
Comparison . . . . .	82
4.5. Model Complexity . . . . .	84
The Nested Model Set . . . . .	84
Occam’s Razor . . . . .	85
4.6. Move-Stop Hypothesis Tracking . . . . .	86
4.7. Simultaneous Multiple Moving Object Tracking . . . . .	87
4.8. Summary . . . . .	88
CHAPTER 5. Data Association . . . . .	89
5.1. Data Association in the Small . . . . .	90

Object Score Function . . . . .	90
Gating . . . . .	91
Kinematic Contribution to Data Association . . . . .	92
Geometric Contribution to Data Association . . . . .	93
Other Contributions to Data Association . . . . .	93
5.2. Data Association in the Cluttered . . . . .	93
5.3. Data Association in the Large . . . . .	95
Covariance Increasing . . . . .	95
Information Exploiting . . . . .	96
Ambiguity Modelling . . . . .	98
5.4. Summary . . . . .	101
CHAPTER 6. Implementation . . . . .	103
6.1. Solving the Moving Object Tracking Problem Locally or Globally? . . . . .	103
6.2. Moving Object Detection . . . . .	105
Consistency-based Detection . . . . .	105
Moving Object Map based Detection . . . . .	107
Iterated SLAM with DATMO . . . . .	107
6.3. Stationary Object Map and Moving Object Map . . . . .	108
6.4. Data-Driven Approach to Non-Linearity and Non-Gaussianity . . . . .	109
6.5. Experimental Results . . . . .	110
Detection and Data Association . . . . .	110
Tracking . . . . .	111
3D ( $2\frac{1}{2}$ D) City-Sized SLAM . . . . .	117
6.6. 2-D Environment Assumption in 3-D Environments . . . . .	119
6.7. Sensor Selection and Limitation . . . . .	123
6.8. Ground Truth . . . . .	125
6.9. Summary . . . . .	127
CHAPTER 7. Conclusion . . . . .	129
7.1. Summary . . . . .	129
7.2. Future Extensions . . . . .	131
Between SLAM with GO and SLAM with DATMO . . . . .	131
Heterogeneous Sensor Fusion . . . . .	131
4-D Environments . . . . .	132
Toward Scene Understanding . . . . .	133
7.3. Conclusion . . . . .	133
APPENDIX A. Notations and Acronyms . . . . .	135
A.1. Notations . . . . .	135
A.2. Acronyms . . . . .	136
BIBLIOGRAPHY . . . . .	137



## LIST OF FIGURES

---

1.1	Robotics for safe driving . . . . .	2
1.2	A traffic scene on a highway . . . . .	4
1.3	A traffic scene in an urban area . . . . .	4
1.4	SLAM vs. DATMO . . . . .	4
1.5	City-sized SLAM . . . . .	6
1.6	Tracking difficulty vs. degrees of freedom . . . . .	9
1.7	The Navlab8 testbed . . . . .	10
1.8	The Navlab11 testbed . . . . .	10
1.9	Raw data from the Navlab11 testbed . . . . .	11
1.10	Result of SLAM with DATMO . . . . .	11
1.11	Thesis overview . . . . .	12
2.1	The SLAM process, the MOT process and the SLAMMOT process . . . . .	16
2.2	Compounding of spatial relationships . . . . .	17
2.3	The inverse relationship . . . . .	18
2.4	The tail-to-tail relationship . . . . .	19
2.5	A Dynamic Bayesian Network (DBN) of the SLAM problem . . . . .	21
2.6	The initialization stage of SLAM . . . . .	22
2.7	A DBN representing the initialization stage of SLAM . . . . .	22
2.8	The prediction stage of SLAM . . . . .	23
2.9	A DBN representing the prediction stage of SLAM . . . . .	23
2.10	The data association stage of SLAM . . . . .	24
2.11	A DBN representing the data association stage of SLAM . . . . .	24
2.12	The update stage of SLAM . . . . .	25
2.13	A DBN representing the update stage of SLAM . . . . .	25
2.14	A DBN for multiple model based moving object tracking . . . . .	28
2.15	The GPB1 algorithm of one cycle for 2 switching models . . . . .	30
2.16	The GPB2 algorithm of one cycle for 2 switching models . . . . .	31
2.17	The IMM algorithm of one cycle for 2 switching models . . . . .	31
2.18	The initialization stage of moving object tracking . . . . .	34
2.19	A DBN representing the initialization stage of moving object tracking . . . . .	34
2.20	The prediction stage of moving object tracking . . . . .	34

LIST OF FIGURES

2.21	A DBN representing the prediction stage of moving object tracking . . .	34
2.22	The data association stage of moving object tracking . . . . .	35
2.23	A DBN representing the data association stage of moving object tracking	35
2.24	The update stage of moving object tracking . . . . .	35
2.25	A DBN representing the update stage of moving object tracking . . . . .	35
2.26	Model Selection . . . . .	36
2.27	Move-stop-move object tracking . . . . .	36
2.28	A DBN for SLAM with Generic Objects . . . . .	37
2.29	A DBN of the SLAM with DATMO problem of duration three with one moving object and one stationary object . . . . .	42
2.30	The initialization stage of SLAM with DATMO . . . . .	42
2.31	A DBN representing the initialization stage of SLAM with DATMO . . .	42
2.32	The prediction stage of SLAM with DATMO . . . . .	43
2.33	A DBN representing the prediction stage of SLAM with DATMO . . . . .	43
2.34	The data association stage of SLAM with DATMO . . . . .	43
2.35	A DBN representing the data association stage of SLAM with DATMO	43
2.36	The update stage of the SLAM part of SLAM with DATMO . . . . .	43
2.37	A DBN representing the update stage of the SLAM part of SLAM with DATMO . . . . .	43
2.38	The update stage of the DATMO part of SLAM with DATMO . . . . .	44
2.39	A DBN representing the update stage of the DATMO part of SLAM with DATMO . . . . .	44
3.1	Vegetation and plant object: Bush . . . . .	47
3.2	Curvy object: A building . . . . .	47
3.3	Circle extraction . . . . .	48
3.4	Line extraction . . . . .	48
3.5	Hierarchical object based representation . . . . .	51
3.6	An example of scan segmentation . . . . .	52
3.7	Footprints of the measurement from SICK LMS 291 . . . . .	53
3.8	SICK LMS 211/221/291 noise model . . . . .	53
3.9	Scan segments of two scans . . . . .	56
3.10	An initial guess of the relative transformation . . . . .	57
3.11	Results of segment 1 registration . . . . .	57
3.12	Results of Figure 3.11 are shown with the whole scans . . . . .	58
3.13	Sampling-based uncertainty estimation . . . . .	58
3.14	The corresponding sample means and covariances using different numbers of samples. . . . .	59
3.15	The occupancy grid map of Segment 1 of <b>A</b> . . . . .	60
3.16	The occupancy grid map of Segment 1 of <b>B</b> . . . . .	60
3.17	The normalized correlations of the samples . . . . .	61
3.18	Mean and covariance estimation by clustering . . . . .	61
3.19	A wide variety of moving objects in urban areas . . . . .	63

3.20	Different portions of a moving car . . . . .	64
3.21	Registration results of the example in Figure 3.20 using the SCRIM algorithm . . . . .	65
3.22	Aerial photo of the CMU neighborhood . . . . .	66
3.23	Pose estimates from the inertial measurement system . . . . .	66
3.24	Generated grid maps along the trajectory . . . . .	67
3.25	Details of the grid maps . . . . .	68
3.26	The data association in the large problem . . . . .	69
3.27	The result without loop-closing . . . . .	70
3.28	The result with loop-closing . . . . .	71
4.1	Model complexity . . . . .	74
4.2	A simulation of the constant acceleration maneuver . . . . .	78
4.3	The velocity estimates and the probabilities of the CV model and the CA model in the constant acceleration maneuver simulation . . . . .	78
4.4	A simulation of the constant velocity maneuver . . . . .	79
4.5	The velocity estimates and the probabilities of the CV model and the CA model in the constant velocity motion simulation . . . . .	79
4.6	A simulation of the turning maneuver . . . . .	80
4.7	The velocity estimates and the probabilities of the CV model and the CA model in the turning maneuver simulation . . . . .	80
4.8	A simulation of the move-stop-move maneuver . . . . .	80
4.9	The velocity estimates and the probabilities of the CV model and the CA model in the move-stop-move maneuver simulation . . . . .	80
4.10	The simulation in which a stationary object is tracked using a Kalman filter with the CV model . . . . .	81
4.11	A simulation in which the stationary object is tracked using a Kalman filter with the stop model . . . . .	83
4.12	A simulation in which a stationary object is tracked using the stationary process model . . . . .	83
4.13	A simulation in which an object moving at a constant velocity is tracked using a Kalman filter with the stop model . . . . .	83
4.14	A simulation in which a constant velocity moving object is tracked using the stationary process model . . . . .	83
4.15	The nested model set . . . . .	84
4.16	Occam’s Razor . . . . .	85
4.17	A simulation of the move-stop-move maneuver tracked by move-stop hypothesis tracking. . . . .	87
4.18	The enlargement of Figure 4.17 . . . . .	87
5.1	Gating . . . . .	92
5.2	Data association in the cluttered . . . . .	94
5.3	Clustering . . . . .	94
5.4	Data association in the large . . . . .	95
5.5	Covariance increasing . . . . .	96

LIST OF FIGURES

5.6	The grid-map pair of the same region built at different times: Grid map 1 and Grid map 16 . . . . .	97
5.7	Recognition and localization results using different scales of grid map 1 and grid map 16 . . . . .	98
5.8	Two sequences. The relative starting locations of these two sequences are assumed to be unknown . . . . .	99
5.9	Details of grid map 1-9 . . . . .	99
5.10	Details of grid map 10-21 . . . . .	100
5.11	The bar graph of the maximum correlation values of the grid map pairs between the grid map 1-14 sequence and the grid map 15-21 sequence .	101
5.12	The slices of Figure 5.11 . . . . .	101
5.13	The total correlation value of the consecutive grid maps . . . . .	101
6.1	Performing detection, data association and motion modelling in a global frame . . . . .	104
6.2	Performing detection, data association and motion modelling with respect to a temporary global coordinate system . . . . .	104
6.3	Case 1 of detection . . . . .	106
6.4	Case 2 of detection . . . . .	106
6.5	Consistency-based detection . . . . .	107
6.6	Iterated SLAM with DATMO . . . . .	108
6.7	Multiple vehicle detection and data association . . . . .	111
6.8	Pedestrian detection and data association . . . . .	112
6.9	Pedestrian detection and data association . . . . .	112
6.10	Bus detection and data association . . . . .	113
6.11	Temporary stationary objects . . . . .	113
6.12	Tracking results of the example in Figure 3.20 . . . . .	114
6.13	Detection and data association results . . . . .	115
6.14	The partial image from the tri-camera system . . . . .	115
6.15	Raw data of 201 scans . . . . .	115
6.16	Results of multiple ground vehicle tracking . . . . .	115
6.17	Speed estimates . . . . .	116
6.18	An intersection. Pedestrians are pointed out by the arrow. . . . .	116
6.19	Visual images from the tri-camera system . . . . .	116
6.20	Raw data of 141 scans . . . . .	117
6.21	Results of multiple pedestrian tracking . . . . .	117
6.22	Speed estimates of object A . . . . .	117
6.23	Probabilities of the CV and CA models of object A . . . . .	117
6.24	Speed estimates of object B . . . . .	117
6.25	Probabilities of the CV and CA models of object B . . . . .	117
6.26	The scene . . . . .	118
6.27	201 raw scans and the robot trajectory . . . . .	118
6.28	The visual image from the tri-camera system . . . . .	118

6.29	The result of move-stop object tracking using IMM with the CV and CA models . . . . .	118
6.30	Speed estimates from IMM . . . . .	119
6.31	The result of tracking using the move-stop hypothesis tracking algorithm	119
6.32	A 3-D map of several street blocks . . . . .	120
6.33	A 3-D model of the Carnegie Museum of Natural History . . . . .	120
6.34	3-D models of buildings on Filmore street . . . . .	121
6.35	3-D models of parked cars in front of the Carnegie Museum of Art . . .	121
6.36	3-D models of trees on S. Bellefield avenue. . . . .	121
6.37	Dramatic changes between consecutive scans due to a sudden start . .	122
6.38	False measurements from a uphill environment . . . . .	122
6.39	The failure mode of the laser scanners . . . . .	124
6.40	The direct sun effect on the regular camera . . . . .	125
6.41	An available digital map . . . . .	126
6.42	The reconstructed map is overlaid on an aerial photo . . . . .	126
6.43	The same intersections shown in Figure 6.41 are overlaid on our reconstructed map . . . . .	126
7.1	Between SLAM with GO and SLAM with DATMO . . . . .	131
7.2	4-D environments . . . . .	132



## LIST OF TABLES

---

3.1	Representation comparison . . . . .	50
3.2	Object saliency score . . . . .	62
4.1	Model Complexity . . . . .	86
6.1	Features of SICK laser scanners . . . . .	110
6.2	Steepness grades of Pittsburgh hills . . . . .	123



# CHAPTER 1

---

## Introduction

*One, a robot may not injure a human being, or through inaction, allow a human being to come to harm;*

*Two, a robot must obey the orders given by human beings except where such orders would conflict with the First Law;*

*Three, a robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.*

– Isaac Asimov (1920 - 1992)

*"I, Robot"*

**S**CENE UNDERSTANDING is a key prerequisite for making a robot truly autonomous. The scene around the robot consists of stationary and/or moving objects. In applications such as planetary exploration and automated mining, the world around the robot consists of stationary objects, and only the robot can change its and the world's states. In applications such as elder care, office automation, security and safe driving, the world is dynamic, consisting of both stationary and moving entities.

Establishing the spatial and temporal relationships among the robot, stationary objects and moving objects in the scene serves as the basis for scene understanding. *Localization* is the process of establishing the spatial relationships between the robot and stationary objects, *mapping* is the process of establishing the spatial relationships among stationary objects, and *moving object tracking* is the process of establishing the spatial and temporal relationships between moving objects and the robot or between moving objects and stationary objects.

Localization, mapping and moving object tracking are difficult because of uncertainty and unobservable states in the real world. For instance, perception sensors such as cameras, radar and laser range finders, and motion sensors such as odometry and inertial measurement units are noisy. For moving object tracking, the intentions, or control inputs, of

the moving objects are unobservable without using extra sensors mounted on the moving objects.

This dissertation is concerned with the problem of how a robot can accomplish localization, mapping and moving object tracking in the real world. We will provide a theoretical framework that integrates all these problems and explain why all these problems should be solved together. We will find algorithms for efficiently and robustly solving this whole problem of simultaneous localization, mapping and moving object tracking (SLAMMOT). We will demonstrate these algorithms with ample experimental results from a ground vehicle at high speeds in crowded urban areas.

It is believed by many that a solution to the simultaneous localization and mapping (SLAM) problem will open up a vast range of potential applications for autonomous robots (Thorpe and Durrant-Whyte, 2001; Christensen, 2002). We believe that a solution to the simultaneous localization, mapping and moving object tracking problem will expand the potential for robotic applications in proximity to human beings. Robots will be able to work not only *for* people but also *with* people. In the next section, we will illustrate the whole problem with an example application, safe driving. See Figure 1.1 for an illustration.

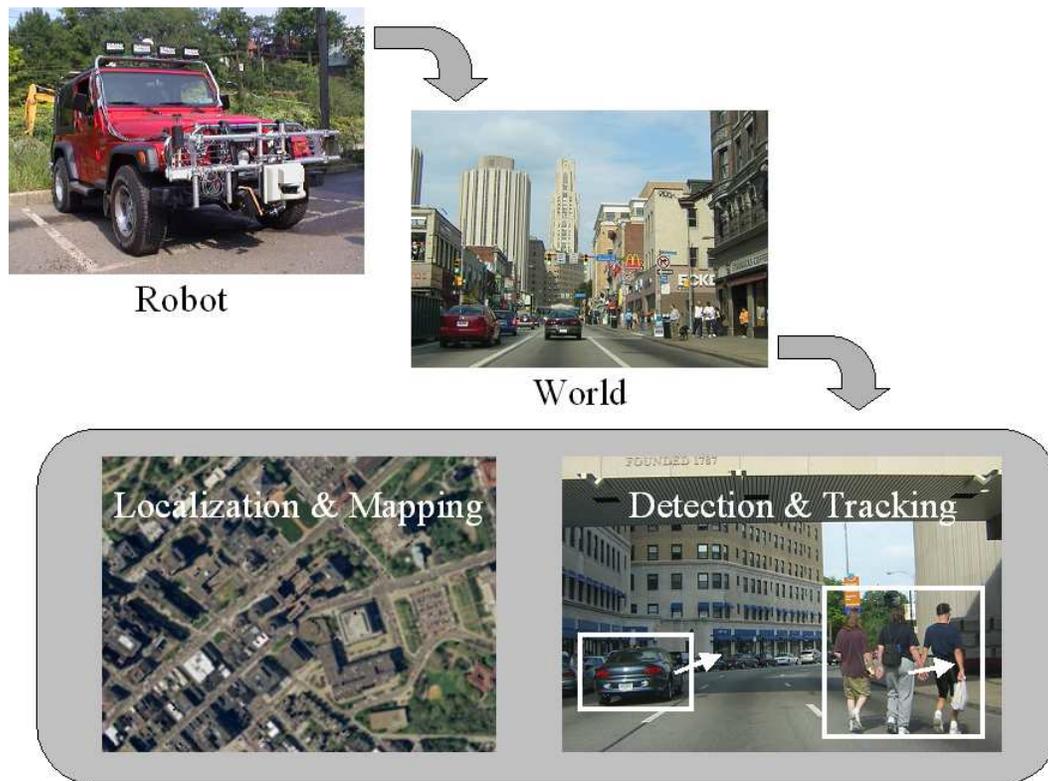


Figure 1.1. Robotics for safe driving. Localization, mapping, and moving object tracking are critical to driving assistance and autonomous driving.

## 1.1. Safe Driving

To improve driving safety and prevent traffic injuries caused by human factors such as speeding and distraction, techniques to understand the surroundings of the vehicle are critical. We believe that being able to detect and track every stationary object and every moving object, to reason about the dynamic traffic scene, to detect and predict every critical situation, and to warn and assist drivers in advance, is essential to prevent these kinds of accidents.

### Localization

In order to detect and track moving objects by using sensors mounted on a moving ground vehicle at high speeds, a precise localization system is essential. It is known that GPS and DGPS often fail in urban areas because of urban canyon effects, and good inertial measurement systems (IMS) are very expensive.

If we can have a stationary object map in advance, the map-based localization techniques such as those proposed by (Olson, 2000), (Fox et al., 1999), and (Dellaert et al., 1999) can be used to increase the accuracy of the pose estimate. Unfortunately, it is difficult to build a usable stationary object map because of temporary stationary objects such as parked cars. Stationary object maps of the same scene built at different times could still be different, which means that we still have to do online map building to update the current stationary object map.

### Simultaneous Localization and Mapping

Simultaneous localization and mapping (SLAM) allows robots to operate in an unknown environment and then incrementally build a map of this environment and concurrently use this map to localize robots themselves. Over the last decade, the SLAM problem has attracted immense attention in the mobile robotics literature (Christensen, 2002), and SLAM techniques are at the core of many successful robot systems (Thrun, 2002). However, (Wang and Thorpe, 2002) have shown that SLAM can perform badly in crowded urban environments because of the static environment assumption. Moving objects have to be detected and filtered out.

### Detection and Tracking of Moving Objects

The detection and tracking of moving objects (DATMO) problem has been extensively studied for several decades (Bar-Shalom and Li, 1988, 1995; Blackman and Popoli, 1999).

Even with precise localization, it is not easy to solve the DATMO problem in crowded urban environments because of a wide variety of targets (Wang et al., 2003a).

When cameras are used to detect moving objects, appearance-based approaches are widely used and moving objects can be detected no matter whether they are moving or not. If laser scanners are used, feature-based approaches are usually the preferred solutions. Both appearance-based and feature-based methods rely on prior knowledge of the targets.

In urban areas, there are many kinds of moving objects such as pedestrians, animals, wheelchairs, bicycles, motorcycles, cars, buses, trucks and trailers. Velocities range from under 5 mph (such as a pedestrian's movement) to 50 mph. Figure 1.2 shows a traffic scene on a highway and Figure 1.3 shows a traffic scene in an urban area. When using laser scanners, the features of moving objects can change significantly from scan to scan. As a result, it is very difficult to define features or appearances for detecting specific objects using laser scanners.



Figure 1.2. A traffic scene on a highway.



Figure 1.3. A traffic scene in an urban area.

### SLAM vs. DATMO

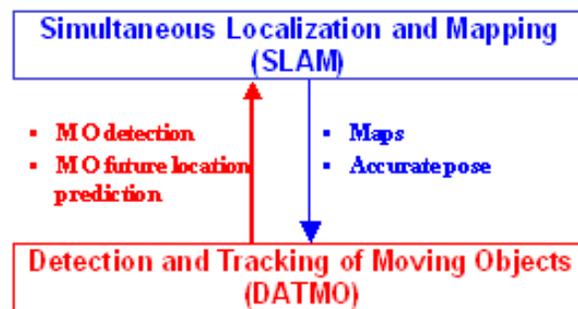


Figure 1.4. SLAM vs. DATMO.

Both SLAM and DATMO have been studied in isolation. However, when driving in crowded urban environments composed of stationary and moving objects, neither of them

is sufficient. The simultaneous localization, mapping and moving object tracking problem aims to tackle the SLAM problem and the DATMO problem at once. Because SLAM provides more accurate pose estimates and a surrounding map, a wide variety of moving objects are detected using the surrounding map without using any predefined features or appearances, and tracking is performed reliably with accurate robot pose estimates. SLAM can be more accurate because moving objects are filtered out of the SLAM process thanks to the moving object location prediction from DATMO. SLAM and DATMO are mutually beneficial. Integrating SLAM with DATMO would satisfy both the safety and navigation demands of safe driving. It would provide a better estimate of the robot's location and information of the dynamic environments, which are critical to driving assistance and autonomous driving.

Although performing SLAM and DATMO at the same time is superior to doing just one or the other, the integrated approach inherits the difficulties and issues from both the SLAM problem and the DATMO problem. Therefore, besides deriving a mathematical formulation to seamlessly integrate SLAM and DATMO, we need to answer the following questions:

- Assuming that the environment is static, can we solve the simultaneous localization and mapping problem from a ground vehicle at high speeds in very large urban environments?
- Assuming that the robot pose estimate is accurate and moving objects are correctly detected, can we solve the moving object tracking problem in crowded urban environments?
- Assuming that the SLAM problem and the DATMO problem can be solved in urban areas, is it feasible to solve the simultaneous localization, mapping and moving object tracking problem? What problems will occur when the SLAM problem and the DATMO problem are solved together?

In the following sections, we will discuss these problems from both theoretical and practical points of view.

## 1.2. City-Sized Simultaneous Localization and Mapping

Since Smith, Self and Cheeseman first introduced the simultaneous localization and mapping (SLAM) problem (Smith and Cheeseman, 1986; Smith et al., 1990), the SLAM problem has attracted immense attention in the mobile robotics literature. SLAM involves simultaneously estimating locations of newly perceived landmarks and the location of the robot itself while incrementally building a map. The web site of the 2002 SLAM summer

school<sup>1</sup> provides a comprehensive coverage of the key topics and state of the art in SLAM. In this section, we address three key issues to accomplish city-sized SLAM: computational complexity, representation, and data association in the large.

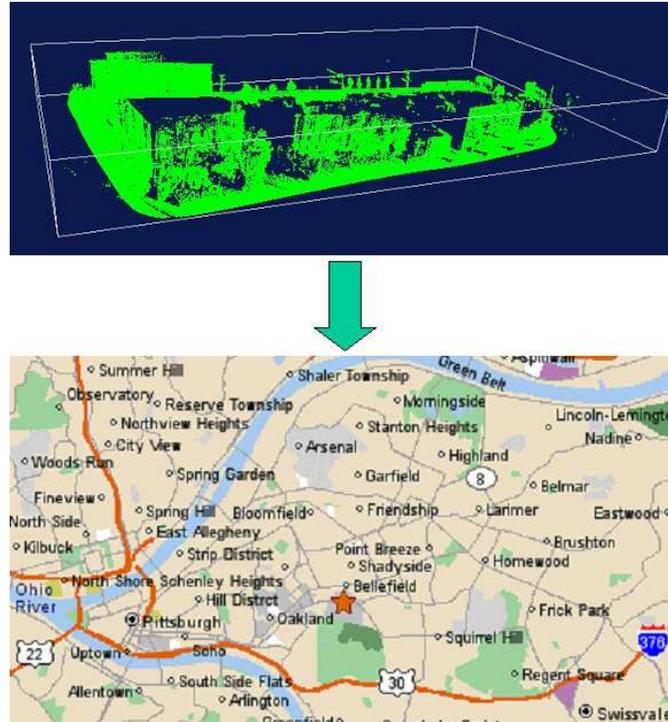


Figure 1.5. City-sized SLAM. Top shows the 3D (2.5D) map of several street blocks using the algorithm addressed in (Wang et al., 2003b). Is it possible to accomplish online SLAM in a city?

### Computational Complexity

In the SLAM literature, it is known that a key bottleneck of the Kalman filter solution is its computational complexity. Because it explicitly represents correlations of all pairs among the robot and stationary objects, both the computation time and memory requirement scale quadratically with the number of stationary objects in the map. This computational burden restricts applications to those in which the map can have no more than a few hundred stationary objects.

Recently, this problem has been subject to intense research. Approaches using approximate inference, using exact inference on tractable approximations of the true model, and using approximate inference on an approximate model have been proposed. In this

<sup>1</sup><http://www.cas.kth.se/SLAM/>

dissertation, we will take advantage of these promising approaches and focus on the representation and data association issues. More details about the computational complexity issue will be addressed in Section 2.2.

### **Representation**

Even with an advanced algorithm to deal with computational complexity, most SLAM applications are still limited to indoor environments (Thrun, 2002) or specific environments and conditions (Guivant et al., 2000) because of significant issues in defining environment representation and identifying an appropriate methodology for fusing data in this representation (Durrant-Whyte, 2001). For instance, feature-based approaches have an elegant solution by using a Kalman filter or an information filter, but it is difficult to extract features robustly and correctly in outdoor environments. Grid-based approaches do not need to extract features, but they do not provide any direct means to estimate and propagate uncertainty and they do not scale well in very large environments.

In Chapter 3, we will address the representation related issues in detail and describe a hierarchical object based representation for overcoming the difficulties of the city-sized SLAM problem.

### **Data Association in the Large**

Given correct data association in the large, or loop detection, SLAM can build a globally consistent map regardless of the size of the map. In order to obtain correct data association in the large, most large scale mapping systems using moving platforms (Zhao and Shibasaki, 2001; Früh and Zakhor, 2003) are equipped with expensive state estimation systems to assure the accuracy of the state estimation. In addition, independent position information from GPS or aerial photos is used to provide global constraints.

Without these aids, the accumulated error of the pose estimate and unmodelled uncertainty in the real world increase the difficulty of loop detection. For dealing with this issue without access to independent position information, our algorithm based on covariance increasing, information exploiting and ambiguity modelling will be presented in Chapter 5.

In this work, we will demonstrate that it is feasible to accomplish city-sized SLAM.

### 1.3. Moving Object Tracking in Crowded Urban Environments

In order to accomplish moving object tracking in crowded urban areas, three key issues have to be solved: detection, data association in the cluttered, and moving object motion modelling.

#### Detection

Recall that detection of ground moving objects using feature- or appearance-based approaches is infeasible because of the wide variety of targets in urban areas. In Chapter 6, the consistency-based detection and the moving object map based detection will be described for robustly detecting moving objects using laser scanners.

#### Cluttered Environments

Urban areas are often cluttered, as illustrated in Figure 1.3. In the tracking literature, there are a number of techniques for solving data association in the cluttered such as multiple hypothesis tracking (MHT) approaches (Reid, 1979; Cox and Hingorani, 1996) and joint probabilistic data association (JPDA) approaches (Fortmann et al., 1983; Schulz et al., 2001).

In addition to the MHT approach, we use geometric information of moving objects to aid data association in the cluttered because of the rich geometric information contained in laser scanner measurements, which will be discussed in Chapter 3 and Chapter 5.

#### Motion Modelling

In SLAM, we can use odometry and the identified robot motion model to predict the future location of the robot, so that the SLAM problem is an *inference* problem. However, in DATMO neither *a priori* knowledge of moving objects' motion models nor odometry measurements about moving objects is available. In practice, motion modes of moving objects are often partially unknown and time-varying. Therefore, the motion modes of the moving object tracking have to be learned online. In other words, moving object tracking is a *learning* problem.

In the tracking literature, multiple model based approaches have been proposed to solve the motion modelling problem. The related approaches will be reviewed in Section 2.3.

Compared to air and marine target tracking, ground moving object tracking (Chong et al., 2000; Shea et al., 2000) is more complex because of more degrees of freedom (e.g., move-stop-move maneuvers). In Chapter 4, we will present a stationary motion model and a move-stop hypothesis tracking algorithm to tackle this issue.

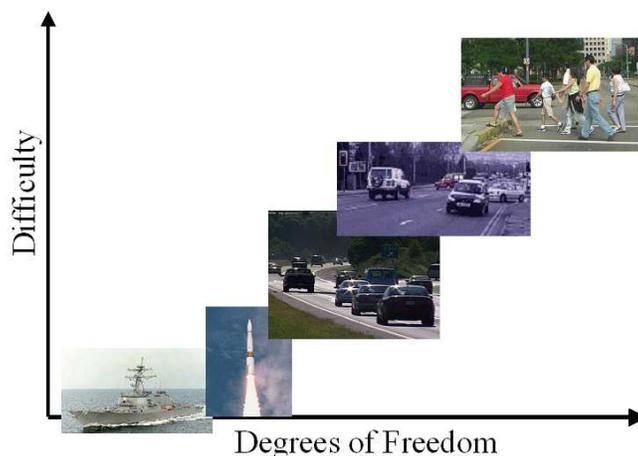


Figure 1.6. Tracking difficulty vs. degrees of freedom. More degrees-of-freedom of a moving object more difficult tracking.

#### 1.4. Simultaneous Localization, Mapping and Moving Object Tracking

After establishing capabilities to solve the SLAM problem and the DATMO problem in urban areas, it is feasible to solve the simultaneous localization, mapping and moving object tracking problem. Because simultaneous localization, mapping and moving object tracking is a more general process based on the integration of SLAM and moving object tracking, it inherits the complexity, data association, representation (perception modelling) and motion modelling issues from the SLAM problem and the DATMO problem. It is clear that the simultaneous localization, mapping and moving object tracking problem is not only an *inference* problem but also a *learning* problem.

In Chapter 2, we will present two approaches and derive the corresponding Bayesian formulas for solving the simultaneous localization, mapping and moving object tracking problem: one is SLAM with Generic Objects, or SLAM with GO, and the other is SLAM with DATMO.

#### 1.5. Experimental Setup

Range sensing is essential in robotics for scene understanding. Range information can be from active range sensors or passive range sensors. (Hebert, 2000) presented a broad review of range sensing technologies for robotic applications. In spite of the different characteristics of these range sensing technologies, the theory presented in Chapter 2 does not limit the usage of specific sensors as long as sensor characteristics are properly modelled.

When using more accurate sensors, inference and learning are more practical and tractable. In order to accomplish simultaneous localization, mapping and moving object tracking from a ground vehicle at high speeds, we mainly focus on issues of using active

ranging sensors. SICK scanners<sup>2</sup> are being used and studied in this work. Data sets collected from the Navlab8 testbed (see Figure 1.7) and the Navlab11 testbed (see Figure 1.8) are used to verify the derived formulas and the developed algorithms. Visual images from the omni-directional camera and the tri-camera system are only for visualization. Figure 1.9 shows a raw data set collected from the Navlab11 testbed. For the purpose of comparison, the result from our algorithms is shown in Figure 1.10 where measurements associated with moving objects are filtered out.



Figure 1.7. Left: the Navlab8 testbed. Right: the SICK PLS100 and the omni-directional camera.



Figure 1.8. Right: the Navlab11 testbed. Left: SICK LMS221, SICK LMS291 and the tri-camera system.

## 1.6. Thesis Statement

Performing localization, mapping and moving object tracking concurrently is superior to doing just one or the other. We will establish a mathematical framework that integrates all, and demonstrate that it is indeed feasible to accomplish simultaneous localization,

<sup>2</sup><http://www.sickoptic.com/>

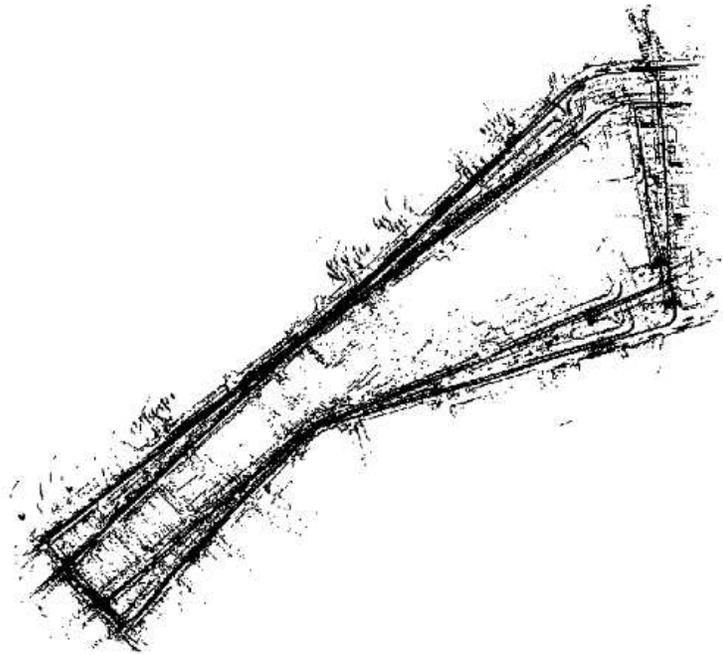


Figure 1.9. Raw data from the Navlab11 testbed. This data set contains  $\sim 36,500$  scans and the travel distance is  $\sim 5$  km.

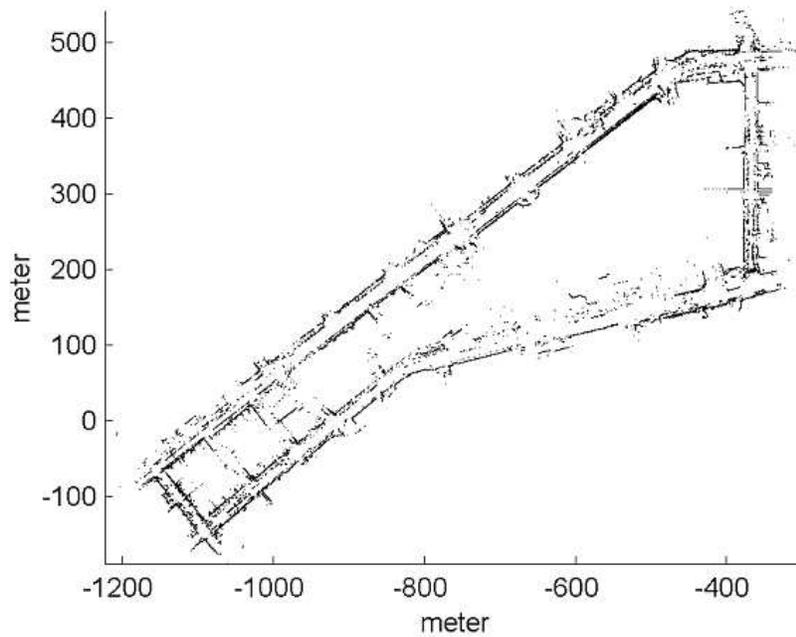


Figure 1.10. Result of SLAM with DATMO. A globally consistent map is generated and measurements associated with moving objects are filtered out.

mapping and moving object tracking from a ground vehicle at high speeds in crowded urban areas.

## 1.7. Document Outline

The organization of this dissertation is summarized in Figure 1.11. We will describe the foundations for solving the SLAMMOT problem in Chapter 2 and the practical issues about perception modelling, motion modelling and data association in the rest of the chapters.

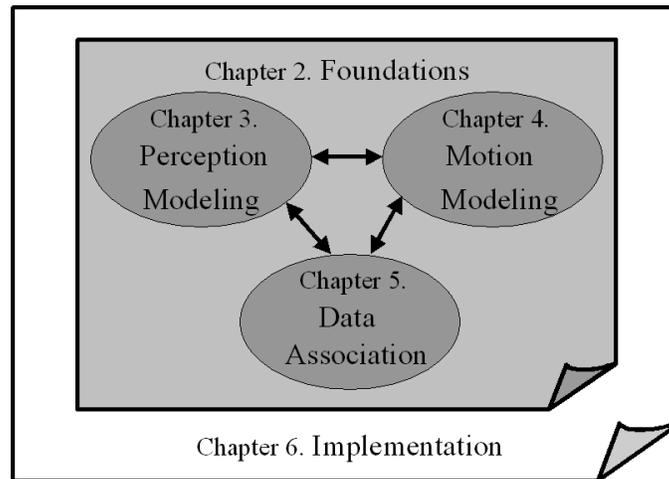


Figure 1.11. Thesis overview.

We begin Chapter 2 with a review of the formulations of the SLAM problem and the moving object tracking problem. We establish a mathematical framework to integrate localization, mapping and moving object tracking, which provides a solid basis for understanding and solving the whole problem. We describe two solutions: SLAM with GO, and SLAM with DATMO. SLAM with GO calculates a joint posterior over all objects (robot pose, stationary objects and moving objects). Such an approach is similar to existing SLAM algorithms, but with additional structure to allow for motion modelling of the moving objects. Unfortunately, it is computationally demanding and infeasible. Consequently, we describe SLAM with DATMO, which is feasible given reliable moving object detection.

In Chapter 3, we address perception modelling issues. We provide a comparison of the main paradigms for perception modelling in terms of uncertainty management, sensor characteristics, environment representability, data compression and loop-closing mechanism. To overcome the limitations of these representation methods and accomplish both SLAM and moving object tracking, we present the hierarchical object-based approach to integrate direct methods, grid-based methods and feature-based methods. When data is uncertain and sparse, the pose estimate from the direct methods such as the iterated closed point (ICP) algorithm may not be correct and the distribution of the pose estimate may not

be described properly. We describe a sampling and correlation based range image matching (SCRIM) algorithm to tackle these issues.

Theoretically, motion modelling is as important as perception modelling in Bayesian approaches. Practically, the performance of tracking strongly relates to motion modelling. In Chapter 4, we address model selection and model complexity issues in moving object motion modelling. A stationary motion model is added to the model set and the move-stop hypothesis tracking algorithm is applied to tackle the move-stop-move or very slow target tracking problem.

In Chapter 5, three data association problems are addressed: data association in the small, data association in the cluttered and data association in the large. We derive formulas to use rich geometric information from perception modelling as well as kinematics from motion modelling for solving data association. Data association in the large, or the revisiting problem, is very difficult because of accumulated pose estimate errors, unmodelled uncertainty, occlusion, and temporary stationary objects. We will demonstrate that following three principles - covariance increasing, information exploiting and ambiguity modelling - is sufficient for robustly detecting loops in very large scale environments.

In Chapter 6, we address the implementation issues for linking foundations, perception modelling, motion modelling and data association together. We provide two practical and reliable algorithms for detecting moving objects using laser scanners. For verifying the theoretical framework and the described algorithms, we show ample results carried out with Navlab8 and Navlab11 at high speeds in crowded urban and suburban areas. We also point out the limitations of our system due to the 2-D environment assumption and sensor failures.

Finally, we conclude with a summary of this work and suggest future extensions in Chapter 7.



## CHAPTER 2

---

### Foundations

*The essence of the Bayesian approach is to provide a mathematical rule explaining how you should change your existing beliefs in the light of new evidence.*

*– In praise of Bayes, the Economist (9/30/00)*

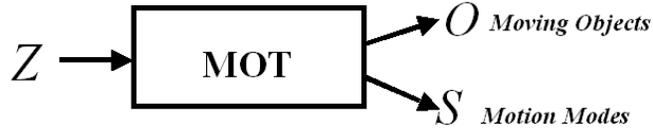
**B**AYESIAN THEORY has been a solid basis for formalizing and solving many statistics, control, machine learning and computer vision problems. The *simultaneous localization, mapping and moving object tracking* problem involves not only accomplishing SLAM in dynamic environments but also detecting and tracking these dynamic objects. Bayesian theory also provides a useful guidance for understanding and solving this problem.

SLAM and moving object tracking can both be treated as processes. SLAM assumes that the surrounding environment is static, containing only stationary objects. The inputs of the SLAM process are measurements from *perception sensors* such as laser scanners and cameras, and measurements from *motion sensors* such as odometry and inertial measurement units. The outputs of the SLAM process are robot pose and a stationary object map (see Figure 2.1.a). Given that the sensor platform is stationary or that a precise pose estimate is available, the inputs of the moving object tracking problem are perception measurements and the outputs are locations of moving objects and their motion modes (see Figure 2.1.b). The simultaneous localization, mapping and moving object tracking problem can also be treated as a process *without* the static environment assumption. The inputs of this process are the same as for the SLAM process, but the outputs are not only the robot pose and the map but also the locations and motion modes of the moving objects (see Figure 2.1.c).

Without considering the perception modelling and data association issues in practice, a key issue of the SLAM problem is *complexity*, and a key issue of the moving object tracking problem is *motion modelling*. Because SLAMMOT inherits the complexity issue from the



(a) the simultaneous localization and mapping (SLAM) process



(b) the moving object tracking (MOT) process



(c) the simultaneous localization, mapping and moving object tracking (SLAMMOT) process

**Figure 2.1.** The SLAM process, the MOT process and the SLAMMOT process.  $Z$  denotes the perception measurements,  $U$  denotes the motion measurements,  $x$  is the true robot state,  $M$  denotes the locations of the stationary objects,  $O$  denotes the states of the moving objects and  $S$  denotes the motion modes of the moving objects.

SLAM problem and the motion modelling issue from the moving object tracking problem, the SLAMMOT problem is not only an *inference* problem but also a *learning* problem.

In this chapter, we first review uncertain spatial relationships which are essential to the SLAM problem, the MOT problem, and the SLAMMOT problem. We will briefly review the Bayesian formulas of the SLAM problem and the moving object tracking problem. In addition, Dynamic Bayesian Networks (DBNs)<sup>1</sup> are used to show the dependencies between the variables of these problems and explain how to compute these formulas. We will present two approaches for solving the simultaneous localization, mapping and moving object tracking problem: SLAM with GO and SLAM with DATMO. For the sake of simplicity, we assume that perception modelling and data association problems are solved and both stationary objects and moving objects can be represented by *point-features*. The details for dealing these issues will be addressed in the following chapters.

<sup>1</sup>For complicated probabilistic problems, computing the Bayesian formula is often computationally intractable. *Graphical models* (Jordan, 2003) provide a natural tool to visualize the dependencies between the variables of the complex problems, and help simplify the Bayesian formula computations by combining simpler parts and ensuring that the system as a whole is still consistent. *Dynamic Bayesian Networks* (DBNs) (Murphy, 2002) are directed graphical models of stochastic processes.

## 2.1. Uncertain Spatial Relationships

For solving the SLAM problem, the MOT problem or the SLAMMOT problem, manipulating uncertain spatial relationships is fundamental. In this section we only intuitively review the spatial relationships for the two dimensional case with three degrees-of-freedom. See (Smith et al., 1990) for a derivation.

### Compounding

In an example in which a moving object is detected by a sonar mounted on a robot, we need to *compound* the uncertainty from the robot pose estimate and the uncertainty from the sonar measurement in order to correctly represent the location of this moving object and the corresponding distribution with respect to the world coordinate system.

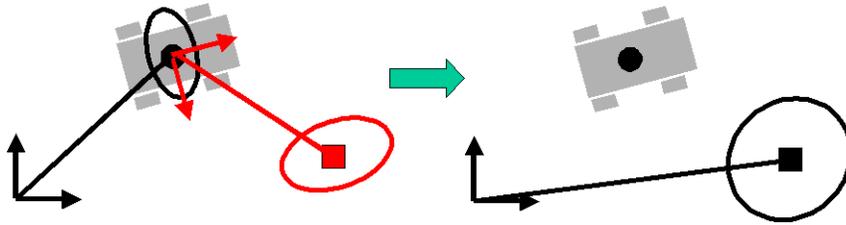


Figure 2.2. Compounding of spatial relationships.

Given two spatial relationships,  $\mathbf{x}_{ij}$  and  $\mathbf{x}_{jk}$ , the formula for compounding  $\mathbf{x}_{ik}$  from  $\mathbf{x}_{ij}$  and  $\mathbf{x}_{jk}$  is:

$$\mathbf{x}_{ik} \triangleq \oplus(\mathbf{x}_{ij}, \mathbf{x}_{jk}) = \begin{bmatrix} x_{jk} \cos \theta_{ij} - y_{jk} \sin \theta_{ij} + x_{ij} \\ x_{jk} \sin \theta_{ij} + y_{jk} \cos \theta_{ij} + y_{ij} \\ \theta_{ij} + \theta_{jk} \end{bmatrix} \quad (2.1)$$

where  $\oplus$  is the compounding operator, and  $\mathbf{x}_{ij}$  and  $\mathbf{x}_{jk}$  are defined by:

$$\mathbf{x}_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \end{bmatrix}, \quad \mathbf{x}_{jk} = \begin{bmatrix} x_{jk} \\ y_{jk} \\ \theta_{jk} \end{bmatrix}$$

Let  $\mu$  be the mean and  $\Sigma$  be the covariance. The first-order estimate of the mean of the compounding operation is:

$$\mu_{\mathbf{x}_{ik}} \approx \oplus(\mu_{\mathbf{x}_{ij}}, \mu_{\mathbf{x}_{jk}}) \quad (2.2)$$

The first order estimate of the covariance is:

$$\Sigma_{\mathbf{x}_{ik}} \approx \nabla_{\oplus} \begin{bmatrix} \Sigma_{\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{ij}\mathbf{x}_{jk}} \\ \Sigma_{\mathbf{x}_{jk}\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{jk}} \end{bmatrix} \nabla_{\oplus}^T \quad (2.3)$$

where the Jacobian of the compounding operation,  $\nabla_{\oplus}$ , is defined by:

$$\nabla_{\oplus} \triangleq \frac{\partial \oplus(\mathbf{x}_{ij}, \mathbf{x}_{jk})}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jk})} = \begin{bmatrix} 1 & 0 & -(y_{ik} - y_{ij}) & \cos \theta_{ij} & -\sin \theta_{ij} & 0 \\ 0 & 1 & (x_{ik} - x_{ij}) & \sin \theta_{ij} & \cos \theta_{ij} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

In the case that the two relationships are independent, we can rewrite the first-order estimate of the covariance as:

$$\Sigma_{\mathbf{x}_{ik}} \approx \nabla_{1\oplus} \Sigma_{\mathbf{x}_{ik}} \nabla_{1\oplus}^T + \nabla_{2\oplus} \Sigma_{\mathbf{x}_{jk}} \nabla_{2\oplus}^T \quad (2.5)$$

where  $\nabla_{1\oplus}$  and  $\nabla_{2\oplus}$  are the left and right halves of the compounding Jacobian. The compounding relationship is also called the *head-to-tail* relationship in (Smith et al., 1990).

### The Inverse Relationship

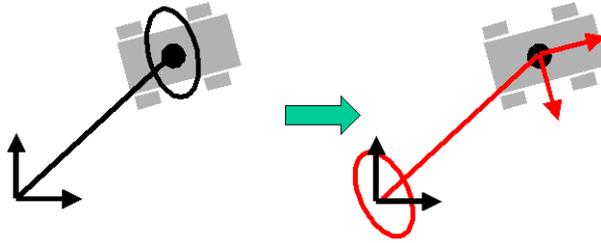


Figure 2.3. The inverse relationship.

Figure 2.3 shows the inverse relationship. For example, given the robot pose in the world coordinate frame,  $\mathbf{x}_{ij}$ , the origin of the world frame with respect to the robot frame,  $\mathbf{x}_{ji}$ , is:

$$\mathbf{x}_{ji} \triangleq \ominus(\mathbf{x}_{ij}) = \begin{bmatrix} -x_{ij} \cos \theta_{ij} - y_{ij} \sin \theta_{ij} \\ x_{ij} \sin \theta_{ij} - y_{ij} \cos \theta_{ij} \\ -\theta_{ij} \end{bmatrix} \quad (2.6)$$

where  $\ominus$  is the inverse operator.

The first-order estimate of the mean of the inverse operation is:

$$\mu_{\mathbf{x}_{ji}} \approx \ominus(\mu_{\mathbf{x}_{ij}})$$

and the first-order covariance estimate is:

$$\Sigma_{\mathbf{x}_{ji}} \approx \nabla_{\ominus} \Sigma_{\mathbf{x}_{ij}} \nabla_{\ominus}^T$$

where the Jacobian for the inverse operation,  $\nabla_{\ominus}$ , is:

$$\nabla_{\ominus} \triangleq \frac{\partial \mathbf{x}_{ji}}{\partial \mathbf{x}_{ij}} = \begin{bmatrix} -\cos \theta_{ij} & -\sin \theta_{ij} & y_{ji} \\ \sin \theta_{ij} & -\cos \theta_{ij} & -x_{ji} \\ 0 & 0 & -1 \end{bmatrix} \quad (2.7)$$

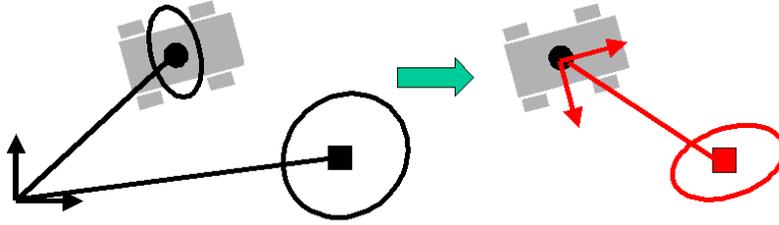


Figure 2.4. The tail-to-tail relationship.

### The Tail-to-Tail Relationship

For local navigation or obstacle avoidance, it is more straightforward to use the locations of moving objects in the robot frame than the locations with respect to the world coordinate system. In the example of Figure 2.4, given the locations of the robot  $\mathbf{x}_{ij}$  and a moving object  $\mathbf{x}_{ik}$  in the world frame, we want to know the location of this moving object,  $\mathbf{x}_{jk}$ , and its distribution,  $\Sigma_{\mathbf{x}_{jk}}$ , in the robot frame, which can be calculated recursively by:

$$\mathbf{x}_{jk} \triangleq \oplus(\ominus(\mathbf{x}_{ij}), \mathbf{x}_{ik}) = \oplus(\mathbf{x}_{ji}, \mathbf{x}_{ik}) \quad (2.8)$$

This relationship is called the *tail-to-tail* relationship in (Smith et al., 1990). The first-order estimate of the mean of this tail-to-tail operation is:

$$\mu_{\mathbf{x}_{jk}} \approx \oplus(\ominus(\mu_{\mathbf{x}_{ij}}), \mu_{\mathbf{x}_{ik}}) \quad (2.9)$$

and the first-order covariance estimate can be computed in a similar way:

$$\Sigma_{\mathbf{x}_{jk}} \approx \nabla_{\oplus} \begin{bmatrix} \Sigma_{\mathbf{x}_{ji}} & \Sigma_{\mathbf{x}_{ji}\mathbf{x}_{jk}} \\ \Sigma_{\mathbf{x}_{jk}\mathbf{x}_{ji}} & \Sigma_{\mathbf{x}_{jk}} \end{bmatrix} \nabla_{\oplus}^T \approx \nabla_{\oplus} \begin{bmatrix} \nabla_{\ominus} \Sigma_{\mathbf{x}_{ij}} \nabla_{\ominus}^T & \Sigma_{\mathbf{x}_{ij}\mathbf{x}_{jk}} \nabla_{\ominus}^T \\ \nabla_{\ominus} \Sigma_{\mathbf{x}_{jk}\mathbf{x}_{ij}} & \Sigma_{\mathbf{x}_{jk}} \end{bmatrix} \nabla_{\oplus}^T \quad (2.10)$$

Note that this tail-to-tail operation is often used in data association and moving object tracking.

### Unscented Transform

As addressed above, these spatial uncertain relationships are non-linear functions and are approximated by their first-order Taylor expansion for estimating the means and the covariances of their outputs. In the cases that the function is not approximately linear in the likely region of its inputs or the Jacobian of the function is unavailable, the *unscented transform* (Julier, 1999) can be used to improve the estimate accuracy. (Wan and van der Merwe, 2000) shows an example of using the unscented transform technique.

## 2.2. Simultaneous Localization and Mapping

In this section, we address the formulation, calculation procedures, computational complexity and practical issues of the SLAM problem.

### Formulation of SLAM

The general formula for the SLAM problem can be formalized in the probabilistic form as:

$$p(x_k, M \mid u_1, u_2, \dots, u_k, z_0, z_1, \dots, z_k) \quad (2.11)$$

where  $x_k$  is the true pose of the robot at time  $k$ ,  $u_k$  is the measurement from motion sensors such as odometry and inertial sensors at time  $k$ ,  $z_k$  is the measurement from perception sensors such as laser scanner and camera at time  $k$ , and  $M$  is stochastic stationary object map which contains  $l$  landmarks,  $m^1, m^2, \dots, m^l$ . In addition, we define the following set to refer data leading up to time  $k$ :

$$Z_k \triangleq \{z_0, z_1, \dots, z_k\} \quad (2.12)$$

$$U_k \triangleq \{u_1, u_2, \dots, u_k\} \quad (2.13)$$

Therefore, equation (2.11) can be rewritten as:

$$p(x_k, M \mid U_k, Z_k) \quad (2.14)$$

Using Bayes' rule and assumptions that the vehicle motion model is Markov and the environment is static, the general recursive Bayesian formula for SLAM can be derived and expressed as: (See (Thrun, 2002; Majumder et al., 2002) for more details.)

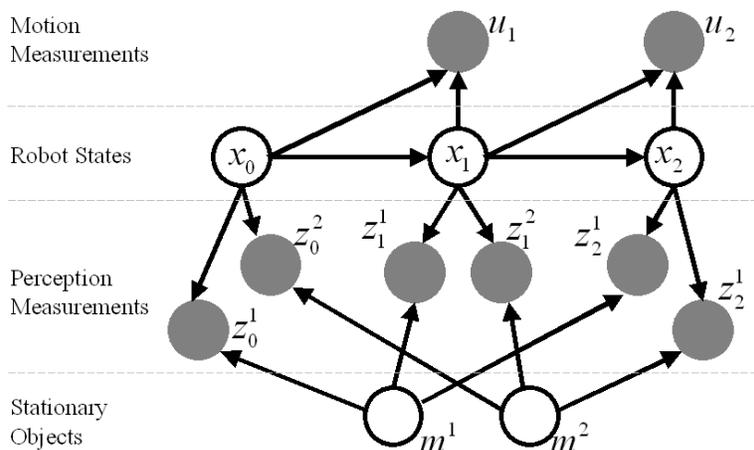
$$\begin{aligned} & \underbrace{p(x_k, M \mid Z_k, U_k)}_{\text{Posterior at } k} \\ & \propto \underbrace{p(z_k \mid x_k, M)}_{\text{Update}} \int \underbrace{p(x_k \mid x_{k-1}, u_k)}_{\text{Prediction}} \underbrace{p(x_{k-1}, M \mid Z_{k-1}, U_{k-1})}_{\text{Posterior at } k-1} dx_{k-1} \end{aligned} \quad (2.15)$$

where  $p(x_{k-1}, M \mid Z_{k-1}, U_{k-1})$  is the posterior probability at time  $k-1$ ,  $p(x_k, M \mid Z_k, U_k)$  is the posterior probability at time  $k$ ,  $p(x_k \mid x_{k-1}, u_k)$  is the motion model, and  $p(z_k \mid x_k, M)$  is the update stage which can be inferred as the perception model.

### Calculation Procedures

Equation 2.15 only explains the computation procedures in each time step but does not address the dependency structure of the SLAM problem. Figure 2.5 shows a Dynamic Bayesian Network of the SLAM problem of duration three, which can be used to visualize the dependencies between the robot and stationary objects in the SLAM problem. In this section, we describe the Kalman filter-based solution of Equation 2.15 with visualization aid from Dynamic Bayesian Networks (Paskin, 2003). The EKF-based framework described

in this section is identical to that used in (Smith and Cheeseman, 1986; Smith et al., 1990; Leonard and Durrant-Whyte, 1991).



**Figure 2.5.** A Dynamic Bayesian Network (DBN) of the SLAM problem of duration three. It shows the dependencies among the motion measurements, the robot, the perception measurements and the stationary objects. In this example, there are two stationary objects,  $m^1$  and  $m^2$ . Clear circles denote hidden continuous nodes and shaded circles denote observed continuous nodes. The edges from stationary objects to measurements are determined by data association. We will walk through this in the next pages.

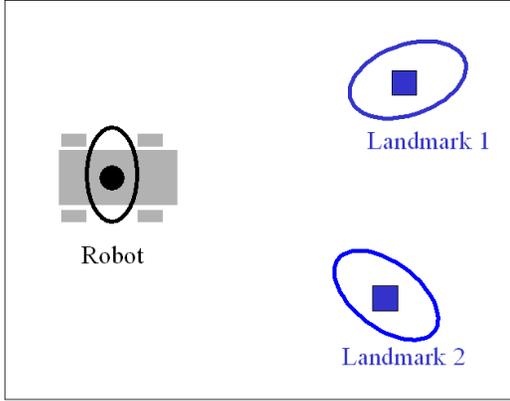
**Stage 1: Initialization.** Figure 2.6 shows the initialization stage, or adding new stationary objects stage. Although the distributions are shown by ellipses in these figures, the Bayesian formula does not assume that the estimations are Gaussian distributions. In this example, two new stationary objects are detected and added to the map. The state  $x_k^S$  of the whole system now is:

$$x_k^S = \begin{bmatrix} x_k \\ m^1 \\ m^2 \end{bmatrix} \quad (2.16)$$

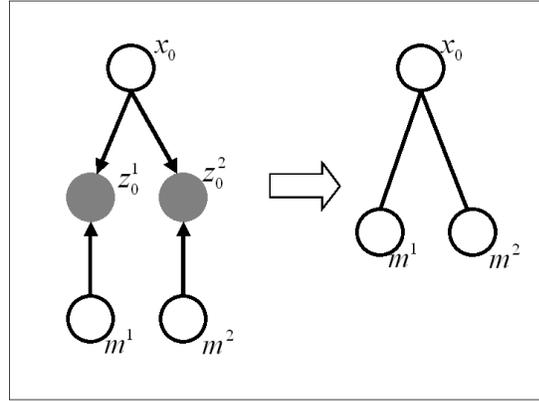
Let the perception model,  $p(z_k | x_k, M)$ , be described as:

$$z_k = h(x_k, M) + w_k \quad (2.17)$$

where  $h$  is the vector-valued perception model and  $w_k \sim \mathcal{N}(0, R_k)$  is the perception error, an uncorrelated zero-mean Gaussian noise sequence with covariance,  $R_k$ . Because the  $z_k$  are the locations of the stationary objects  $M$  with respect to the robot coordinate system, the perception model  $h$  is simply the *tail-to-tail* relationship of the robot and the map. Let the perception sensor return the mean location,  $\hat{z}_0^1$ , and variance,  $R_0^1$ , of the stationary object  $m^1$  and  $\hat{z}_0^2$  and  $R_0^2$  of  $m^2$ . To add these measurements to the map, these measurements are *compounded* with the robot state estimate and its distribution because these measurements



**Figure 2.6.** The initialization stage of SLAM. Solid squares denote stationary objects and black solid circle denotes the robot. Distributions are shown by ellipses.



**Figure 2.7.** A DBN representing the initialization stage of SLAM. After this stage, the undirected graphical model is produced in which two stationary objects and the robot state are directly dependent.

are with respect to the robot coordinate system. Therefore, the mean and covariance of the whole system can be computed as in:

$$\mu_{x_0^S} = \begin{bmatrix} \mu_{x_0} \\ \oplus(\mu_{x_0}, \hat{z}_0^1) \\ \oplus(\mu_{x_0}, \hat{z}_0^2) \end{bmatrix} \quad (2.18)$$

$$\begin{aligned} \Sigma_{x_0^S} &= \begin{bmatrix} \Sigma_{x_0 x_0} & \Sigma_{x_0 m^1} & \Sigma_{x_0 m^2} \\ \Sigma_{x_0 m^1}^T & \Sigma_{m^1 m^1} & \Sigma_{m^1 m^2} \\ \Sigma_{x_0 m^2}^T & \Sigma_{m^1 m^2}^T & \Sigma_{m^2 m^2} \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_{x_0 x_0} & \Sigma_{x_0 x_0} \nabla_{1\oplus}^T & \Sigma_{x_0 x_0} \nabla_{1\oplus}^T \\ \nabla_{1\oplus} \Sigma_{x_0 x_0} & \nabla_{1\oplus} \Sigma_{x_0 x_0} \nabla_{1\oplus}^T + \nabla_{2\oplus} R_0^1 \nabla_{2\oplus}^T & \Sigma_{x_0 x_0} \nabla_{1\oplus}^T \\ \nabla_{1\oplus} \Sigma_{x_0 x_0} & \mathbf{0} & \nabla_{1\oplus} \Sigma_{x_0 x_0} \nabla_{1\oplus}^T + \nabla_{2\oplus} R_0^2 \nabla_{2\oplus}^T \end{bmatrix} \quad (2.19) \end{aligned}$$

This stage is shown as  $p(x_{k-1}, M \mid Z_{k-1}, U_{k-1})$  in equation (2.15). Figure 2.7 shows a DBN representing the initialization stage, or the adding new stationary objects stage, in which the undirected graphical model is produced by moralizing<sup>2</sup> the directed graphical model. The observed nodes are eliminated to produce the final graphical model which shows that two stationary objects and the robot state are directly dependent.

**Stage 2: Predication.** In Figure 2.8, the robot moves and gets a motion measurement  $u_1$  from odometry or inertial sensors. Let the robot motion model,  $p(x_k \mid x_{k-1}, u_k)$ , be

<sup>2</sup>In the *Graphical Model* literature, *moralizing* means adding links between unmarried parents who share a common child.

described as:

$$x_k = f(x_{k-1}, u_k) + v_k \quad (2.20)$$

where  $f(\cdot)$  is the vector of non-linear state transition functions and  $v_k$  is the motion noise, an uncorrelated zero-mean Gaussian noise sequence with covariance,  $Q_k$ . Assuming that the relative motion in the robot frame is given by  $u_k$ , clearly the new location of the robot is the compounding relationship of the robot pose  $x_{k-1}$  and  $u_k$ . Because only the robot moves, only the elements of the mean and the covariance matrix that corresponding to  $x_k$  must be computed. In this example, the mean and the covariance matrix of the whole system can be computed as:

$$\mu_{x_1^S} = \begin{bmatrix} \oplus(\mu_{x_0}, u_1) \\ \mu_{m^1} \\ \mu_{m^2} \end{bmatrix} \quad (2.21)$$

and

$$\begin{aligned} \Sigma_{x_1^S} &= \begin{bmatrix} \Sigma_{x_1 x_1} & \Sigma_{x_1 m^1} & \Sigma_{x_1 m^2} \\ \Sigma_{x_1 m^1}^T & \Sigma_{m^1 m^1} & \Sigma_{m^1 m^2} \\ \Sigma_{x_1 m^2}^T & \Sigma_{m^1 m^2}^T & \Sigma_{m^2 m^2} \end{bmatrix} \\ &= \begin{bmatrix} \nabla_{1 \oplus} \Sigma_{x_0 x_0} \nabla_{1 \oplus}^T + \nabla_{2 \oplus} Q_1 \nabla_{2 \oplus}^T & \nabla_{1 \oplus} \Sigma_{x_0 m^1} & \nabla_{1 \oplus} \Sigma_{x_0 m^2} \\ \Sigma_{x_0 m^1}^T \nabla_{1 \oplus}^T & \Sigma_{m^1 m^1} & \Sigma_{m^1 m^2} \\ \Sigma_{x_0 m^2}^T \nabla_{1 \oplus}^T & \Sigma_{m^1 m^2}^T & \Sigma_{m^2 m^2} \end{bmatrix} \end{aligned} \quad (2.22)$$

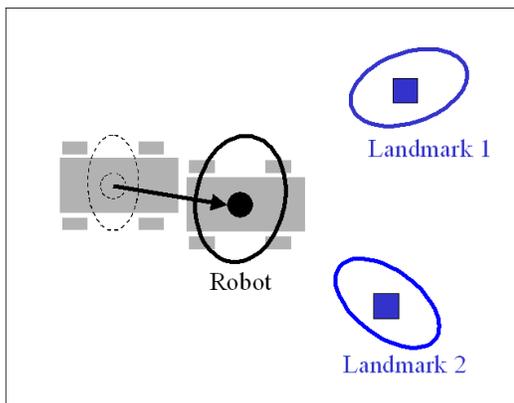


Figure 2.8. The prediction stage of SLAM.

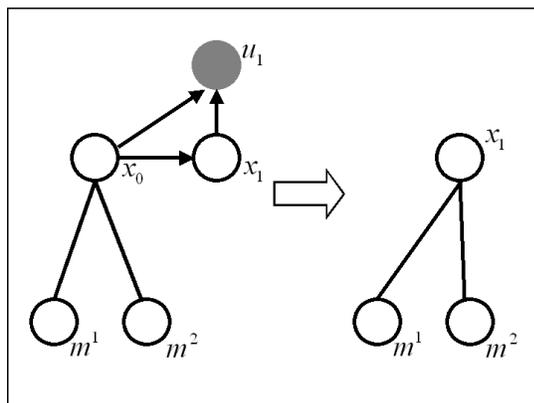


Figure 2.9. A DBN representing the prediction stage of SLAM.

This is the prediction stage of the SLAM problem which is shown as  $\int p(x_k | x_{k-1}, u_k) p(x_{k-1}, M | Z_{k-1}, U_{k-1}) dx_{k-1}$  in equation (2.15). Figure 2.9 shows a DBN representing the prediction stage of the SLAM problem. The new nodes,  $x_1$  and  $u_1$ , are added to the graphical model from the initialization stage. After moralizing the directed graphical model, eliminating the odometry node  $u_1$  and eliminating the node  $x_0$ , the resulting undirected

graphical model is produced in which two stationary objects and the robot state are still directly dependent.

**Stage 3: Data Association.** Figure 2.10 shows that the robot gets new measurements,  $z_1^1$  and  $z_1^2$ , at the new location  $x_1$  and associates  $z_1^1$  and  $z_1^2$  with the stationary object map. This is the data association stage of the SLAM problem. Gating is one of the data association techniques for determining whether a measurement  $z$  originates from some landmark  $m$ . More details about data association will be addressed in Chapter 5.

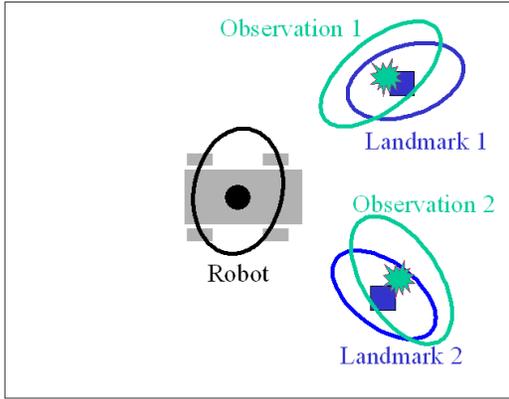


Figure 2.10. The data association stage of SLAM. Irregular stars denote new measurements.

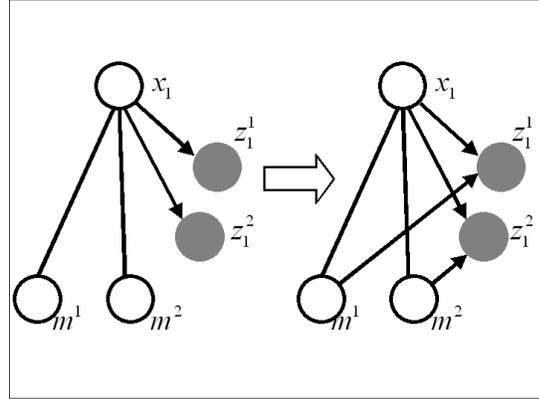


Figure 2.11. A DBN representing the data association stage of SLAM.

Figure 2.11 shows a DBN representing the data association stage. The new perception measurement nodes,  $z_1^1$  and  $z_1^2$ , are added to the graphical model from the prediction stage. After data association, two directed edges are added to connect new measurements with the stationary object map.

**Stage 4: Update.** Figure 2.12 shows the update stage of the SLAM problem. Let the perception sensor return the mean location,  $\hat{z}_1^1$ , and variance,  $R_1^1$ , of the stationary object  $m^1$  and  $\hat{z}_1^2$  and  $R_1^2$  of  $m^2$ . These constraints are used to update the estimate and the corresponding distribution of the whole system with Kalman filtering or other filtering techniques.

An innovation and its corresponding innovation covariance matrix are calculated by:

$$\nu_1 = \mathbf{z}_1 - \hat{\mathbf{z}}_1 \quad (2.23)$$

$$\Sigma_{\nu_1} = \nabla_h \Sigma_{\mathbf{x}_1^s} \nabla_h^T + \Sigma_{R_1} \quad (2.24)$$

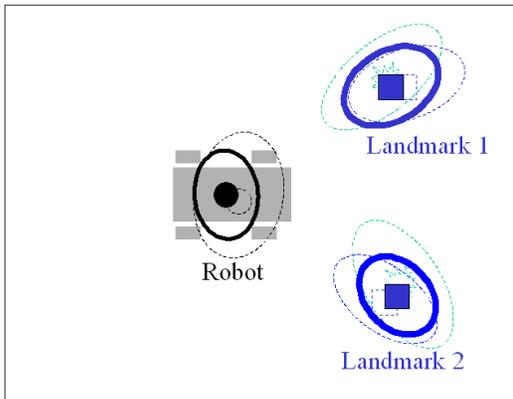


Figure 2.12. The update stage of SLAM

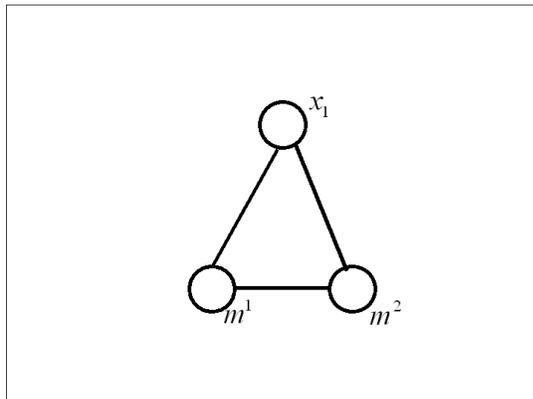


Figure 2.13. A DBN representing the update stage of SLAM

where  $\mathbf{z}_1$  and  $\hat{\mathbf{z}}_1$  are computed by the compounding operation:

$$\mathbf{z}_1 = \begin{bmatrix} \oplus(\ominus(\mu_{x_1}), \mu_{m^1}) \\ \oplus(\ominus(\mu_{x_1}), \mu_{m^2}) \end{bmatrix} \quad (2.25)$$

$$\hat{\mathbf{z}}_1 = \begin{bmatrix} \hat{z}_1^1 \\ \hat{z}_1^2 \end{bmatrix} \quad (2.26)$$

and  $\nabla_h$  is the Jacobian of  $h$  taken at  $\mu_{x_1}$ . Then the state estimate and its corresponding state estimate covariance are updated according to:

$$\mathbf{x}_1^S = \mathbf{x}_1^S + K_1 \nu_1 \quad (2.27)$$

$$\Sigma_{\mathbf{x}_1^S} = \Sigma_{\mathbf{x}_1^S} - K_1 \nabla_h \Sigma_{\mathbf{x}_1^S} \quad (2.28)$$

where the gain matrix is given by:

$$K_1 = \Sigma_{\mathbf{x}_1^S} \nabla_h^T \Sigma_{\nu_1}^{-1} \quad (2.29)$$

This is the update stage of the SLAM problem which is shown as  $p(z_k | x_k, M)$  in equation (2.15). Figure 2.13 shows a DBN representing the update stage of the SLAM problem. After the update stage, the robot and two stationary objects are *fully* correlated.

### Computational Complexity

The Kalman filter solution of the SLAM problem is elegant, but a key bottleneck is its computational complexity. Because it explicitly represents correlations of all pairs among the robot and stationary objects, the size of the covariance matrix of the whole system grows as  $O(l^2)$ , given that the number of stationary objects is  $l$ . The time complexity of the standard EKF operation in the *update* stage is also  $O(l^2)$ . This computational burden restricts applications to those in which the map can have no more than a few hundred

stationary objects. The only way to avoid this quadratically increasing computational requirement is to develop *suboptimal* and *approximate* techniques. Recently, this problem has been subject to intense research. Approaches using approximate inference, using exact inference on tractable approximations of the true model, and using approximate inference on an approximate model have been proposed. These approaches include:

- Thin junction tree filters (Paskin, 2003).
- Sparse extended information filters (Thrun et al., 2002; Thrun and Liu, 2003).
- Submap-based approaches: the *Atlas* framework (Bosse et al., 2003), compressed filter (Guivant and Nebot, 2001) and Decoupled Stochastic Mapping (Leonard and Feder, 1999).
- Rao-Blackwellised particle filters (Montemerlo, 2003).

This topic is beyond the scope intended by this dissertation. (Paskin, 2003) includes an excellent comparison of these techniques.

### Perception Modelling and Data Association

Besides the computational complexity issue, the problems of perception modelling and data association have to be solved in order to accomplish city-sized SLAM. For instance, the described feature-based formulas may not be feasible because extracting features robustly is very difficult in outdoor, urban environments. Data association is difficult in practice because of featureless areas, occlusion, etc. We will address perception modelling in Chapter 3 and data association in Chapter 5.

### 2.3. Moving Object Tracking

Just as with the SLAM problem, the moving object tracking problem can be solved with the mechanism of Bayesian approaches such as Kalman filtering. Assuming correct data association, the moving object tracking problem is easier than the SLAM problem in terms of computational complexity. However, motion models of moving objects are often partially unknown and time-varying. The moving object tracking problem is more difficult than the SLAM problem in terms of online motion model learning. In this section, we address the formulation, mode learning with state inference, calculation procedures and motion modelling issues of the moving object tracking problem.

### Formulation of Moving Object Tracking

The robot (sensor platform) is assumed to be stationary for the sake of simplicity. The general formula for the moving object tracking problem can be formalized in the probabilistic form as:

$$p(o_k, s_k | Z_k) \quad (2.30)$$

where  $o_k$  is the true state of the moving object at time  $k$ , and  $s_k$  is the *true motion mode* of the moving object at time  $k$ , and  $Z_k$  is the perception measurement set leading up to time  $k$ .

Using Bayes' rule, Equation 2.30 can be rewritten as:

$$p(o_k, s_k | Z_k) = p(o_k | s_k, Z_k)p(s_k | Z_k) \quad (2.31)$$

which indicates that the whole moving object tracking problem can be solved by two stages: the first stage is the *mode learning* stage  $p(s_k | Z_k)$ , and the second stage is the *state inference* stage  $p(o_k | s_k, Z_k)$ .

### Mode Learning and State Inference

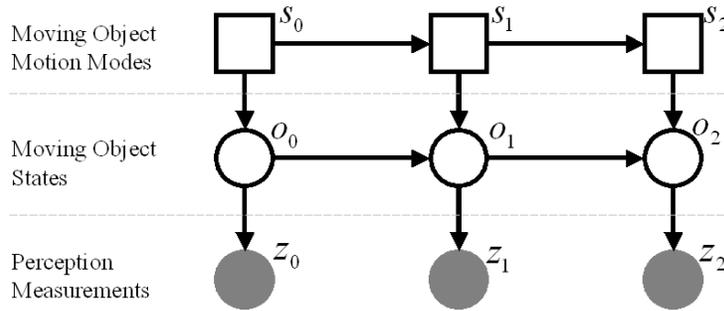
Without *a priori* information, online mode learning of time-series data is a daunting task. In the control literature, specific data collection procedures are designed for *identification* of structural parameters of the system. However, online collected data is often not enough for online identification of the structural parameters in moving object tracking applications.

Fortunately, the motion mode of moving objects can be approximately composed of several motion models such as the constant velocity model, the constant acceleration model and the turning model. Therefore the mode learning problem can be simplified to a *model selection* problem. It is still difficult though because the motion mode of moving objects can be time-varying. In this section, practical *multiple model approaches* are briefly reviewed such as the generalized pseudo-Bayesian (GPB) approaches and the interacting multiple model (IMM) approach. Because the IMM algorithm is integrated into our whole algorithm, the derivation of the IMM algorithm will be described in detail. The multiple model approaches described in this section are identical to those used in (Bar-Shalom and Li, 1988, 1995).

The same problems are solved with *switching dynamic models* in the machine learning literature (Ueda and Ghahramani, 2002; Pavlovic et al., 1999; Ghahramani and Hinton,

1998). In the cases that the models in the model set are linear, such systems are called *jump-linear systems* or *switching linear dynamic models*. However, most of them are batch so that they are not suitable for our applications.

**Fixed Structure Multiple Model Approach for Switching Modes.** In the fixed structure multiple model approach, it is assumed that the mode of the system obeys one of a finite number of models in which the system has both *continuous* nodes as well as *discrete* nodes. Figure 2.14 shows a Dynamic Bayesian Network representing three time steps of an example multiple model approach for solving the moving object tracking problem.



**Figure 2.14.** A DBN for multiple model based moving object tracking. Clear circles denote hidden continuous nodes, clear squares denotes hidden discrete nodes and shaded circles denotes continuous nodes.

The mode of the moving object is assumed to be one of  $r$  possible models which is described by:

$$s_k \in \{\mathcal{M}^j\}_{j=1}^r \quad (2.32)$$

where  $\mathcal{M}$  is the model set.

In practice the system does not always stay in one mode. Because *mode jump* or *mode switch* does occur, the mode-history of the system should be estimated. The mode history through time  $k$  is denoted as  $S_k$

$$S_k = \{s_1, s_2, \dots, s_k\} \quad (2.33)$$

Given  $r$  possible models, the number of possible histories,  $M_k^l$ , is  $r^k$  at time  $k$ , which increases *exponentially* with time. Let  $l$  be the index of the mode history.

$$l = 1, 2, \dots, r^k \quad (2.34)$$

The  $l$ th mode history, or sequence of modes, through time  $k$  is denoted as:

$$\begin{aligned} M_k^l &= \{\mathcal{M}_1^{l_1}, \mathcal{M}_2^{l_2}, \dots, \mathcal{M}_k^{l_k}\} \\ &= \{M_{k-1}^l, \mathcal{M}_k^{l_k}\} \end{aligned} \quad (2.35)$$

where  $l_i$  is the model index at time  $i$  from the history  $l$  and

$$1 \leq l_i \leq r \quad i = 1, \dots, k \quad (2.36)$$

Using Bayes' rule, the conditional probability of the  $l$ th mode history  $M_k^l$  can be obtained as:

$$\begin{aligned} \mu_k^l &\triangleq p(M_k^l | Z_k) \\ &= p(M_k^l | Z_{k-1}, z_k) \\ &= \frac{p(z_k | M_k^l, Z_{k-1})p(M_k^l | Z_{k-1})}{p(z_k | Z_{k-1})} \\ &= \eta \cdot p(z_k | M_k^l, Z_{k-1})p(M_k^l | Z_{k-1}) \\ &= \eta \cdot p(z_k | M_k^l, Z_{k-1})p(\mathcal{M}_k^{l_k}, M_{k-1}^l | Z_{k-1}) \\ &= \eta \cdot p(z_k | M_k^l, Z_{k-1})p(\mathcal{M}_k^{l_k} | M_{k-1}^l, Z_{k-1})\mu_{k-1}^l \end{aligned} \quad (2.37)$$

It is assumed that the mode jump process is a Markov process in which the current node depends only on the previous one.

$$\begin{aligned} p(\mathcal{M}_k^{l_k} | M_{k-1}^l, Z_{k-1}) &= p(\mathcal{M}_k^{l_k} | M_{k-1}^l) \\ &= p(\mathcal{M}_k^{l_k} | \mathcal{M}_k^{l_{k-1}}) \end{aligned} \quad (2.38)$$

Equation 2.37 can be rewritten as:

$$\mu_k^l = \eta p(z_k | M_k^l, Z_{k-1})p(\mathcal{M}_k^{l_k} | \mathcal{M}_k^{l_{k-1}})\mu_{k-1}^l \quad (2.39)$$

in which conditioning on the entire past history is needed even using the assumption that the mode jump process is a Markov process.

Using the total probability theorem, Equation 2.31 can be obtained by:

$$\begin{aligned} p(o_k | Z_k) &= \sum_{l=1}^{r^k} p(o_k | M_k^l, Z_k)p(M_k^l | Z_k) \\ &= \sum_{l=1}^{r^k} p(o_k | M_k^l, Z_k)\mu_k^l \end{aligned} \quad (2.40)$$

This method is not practical because an exponentially increasing number of filters are needed to estimate the state. Also even if the modes are Markov, conditioning on the entire past history is needed. In the same way as dealing with the computational complexity of the SLAM problem, the only way to avoid the exponentially increasing number of histories is to use approximate and suboptimal approaches which merge or reduce the number of the mode history hypotheses in order to make computation tractable.

**The Generalized Pseudo-Bayesian Approaches.** The generalized pseudo-Bayesian (GPB) approaches (Tugnait, 1982) apply a simple suboptimal technique which keeps the histories of the largest probabilities, discards the rest, and renormalizes the probabilities.

In the generalized pseudo-Bayesian approaches of the first order (GPB1), the state estimate at time  $k$  is computed under each possible *current* model. At the end of each cycle, the  $r$  hypotheses are merged into a single hypothesis. Equation 2.40 is simplified as:

$$\begin{aligned}
 p(o_k | Z_k) &= \sum_{j=1}^r p(o_k | \mathcal{M}^j, Z_k) p(\mathcal{M}^j | Z_k) \\
 &= \sum_{j=1}^r p(o_k | \mathcal{M}^j, z_k, Z_{k-1}) \mu_k^j \\
 &\approx \sum_{j=1}^r p(o_k | \mathcal{M}^j, z_k, \hat{o}_{k-1}, \Sigma_{o_{k-1}}) \mu_k^j \quad (2.41)
 \end{aligned}$$

where the  $Z_{k-1}$  is approximately summarized by  $\hat{o}_{k-1}$  and  $\Sigma_{o_{k-1}}$ . The GPB1 approach uses  $r$  filters to produce 1 state estimate. Figure 2.15 describes the GPB1 algorithm.

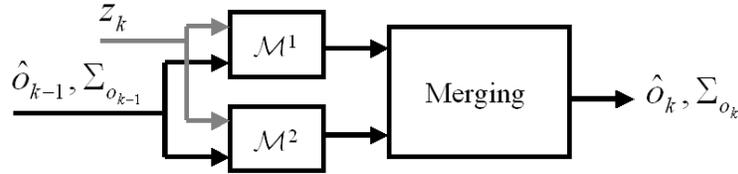


Figure 2.15. The GPB1 algorithm of one cycle for 2 switching models.

In the generalized pseudo-Bayesian approaches of second order (GPB2), the state estimate is computed under each possible model at *current* time  $k$  and *previous* time  $k - 1$ .

$$p(o_k | Z_k) = \sum_{j=1}^r \sum_{i=1}^r p(o_k | \mathcal{M}_k^j, \mathcal{M}_{k-1}^i, Z_k) p(\mathcal{M}_{k-1}^i | \mathcal{M}_k^j, Z_k) p(\mathcal{M}_k^j | Z_k) \quad (2.42)$$

In the GPB2 approach, there are  $r$  estimates and covariances at time  $k - 1$ . Each is predicted to time  $k$  and updated at time  $k$  under  $r$  hypotheses. After the update stage, the  $r^2$  hypotheses are merged into  $r$  at the end of each estimation cycle. The GPB2 approach uses  $r^2$  filters to produce  $r$  state estimates. Figure 2.16 describes the GPB2 algorithm, which does not show the state estimate and covariance combination stage. For output only, the latest state estimate and covariance can be combined from  $r$  state estimates and covariances.

**The Interacting Multiple Model Algorithm.** In the interacting multiple model (IMM) approach (Blom and Bar-Shalom, 1988), the state estimate at time  $k$  is computed under each possible current model using  $r$  filters and each filter uses a suitable mixing of the previous model-conditioned estimate as the initial condition. It has been shown that the

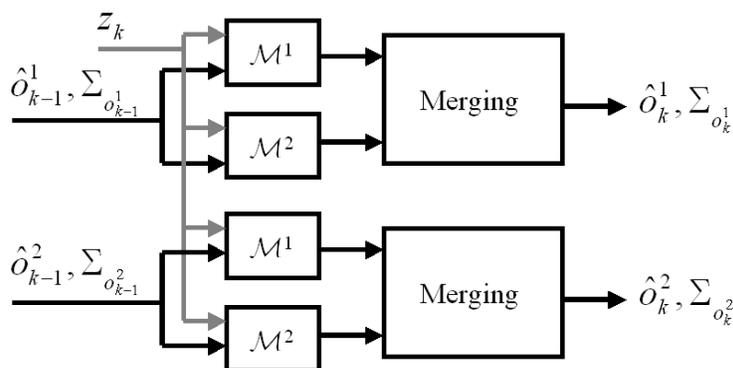


Figure 2.16. The GPB2 algorithm of one cycle for 2 switching models

IMM approach performs significantly better than the GPB1 algorithm and almost as well as the GPB2 algorithm in practice. Instead of using  $r^2$  filters to produce  $r$  state estimates in GPB2, the IMM uses only  $r$  filters to produce  $r$  state estimates. Figure 2.17 describes the IMM algorithm, which does not show the state estimate and covariance combination stage. The derivation of the IMM algorithm is described as the following:

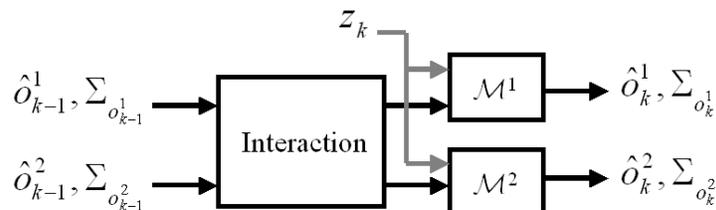


Figure 2.17. The IMM algorithm of one cycle for 2 switching models

Similar to Equation 2.40, the Bayesian formula of the IMM-based tracking problem is described as:

$$\begin{aligned}
 p(o_k | Z_k) &\stackrel{\text{Total Prob.}}{=} \sum_{j=1}^r p(o_k | \mathcal{M}_k^j, Z_k) p(\mathcal{M}_k^j | Z_k) \\
 &\stackrel{\text{Bayes}}{=} \sum_{j=1}^r \frac{p(z_k | o_k, \mathcal{M}_k^j, Z_{k-1}) p(o_k | \mathcal{M}_k^j, Z_{k-1})}{p(z_k | \mathcal{M}_k^j, Z_{k-1})} p(\mathcal{M}_k^j | Z_k) \\
 &= \eta \sum_{j=1}^r p(z_k | o_k, \mathcal{M}_k^j, Z_{k-1}) p(o_k | \mathcal{M}_k^j, Z_{k-1}) p(\mathcal{M}_k^j | Z_k) \\
 &\stackrel{\text{Markov}}{=} \eta \sum_{j=1}^r \underbrace{p(z_k | o_k, \mathcal{M}_k^j)}_{\text{Update}} \underbrace{p(o_k | \mathcal{M}_k^j, Z_{k-1})}_{\text{Prediction}} \underbrace{p(\mathcal{M}_k^j | Z_k)}_{\text{Weighting}} \quad (2.43)
 \end{aligned}$$

where  $p(\mathcal{M}_k^j | Z_k)$  is the *model probability* and can be treated as the weighting of the estimate from the model  $\mathcal{M}_k^j$ .  $p(o_k | \mathcal{M}_k^j, Z_{k-1})$  is the prediction stage and  $p(z_k | o_k, \mathcal{M}_k^j)$  is the update stage. The final estimate is the combination of the estimates from all models.

The model probability,  $p(\mathcal{M}_k^j | Z_k)$ , can be calculated recursively as follows:

$$\begin{aligned} \mu_k^j &\stackrel{\Delta}{=} p(\mathcal{M}_k^j | Z_k) \\ &\stackrel{Bayes}{=} \eta p(z_k | \mathcal{M}_k^j, Z_{k-1}) p(\mathcal{M}_k^j | Z_{k-1}) \\ &\stackrel{Total Prob.}{=} \underbrace{\eta p(z_k | \mathcal{M}_k^j, Z_{k-1})}_{\text{Mode Match}} \sum_{i=1}^r \underbrace{p(\mathcal{M}_k^j | \mathcal{M}_{k-1}^i, Z_{k-1})}_{\text{Mode Transition}} \underbrace{p(\mathcal{M}_{k-1}^i | Z_{k-1})}_{\mu_{k-1}^i} \end{aligned} \quad (2.44)$$

The last term on the right hand side is the model probability of the model  $\mathcal{M}^i$  at time  $k-1$ . The second term on the right hand side is the *mode transition probability*. Here it is assumed that the mode jump process is a Markov process with known mode transition probabilities. Therefore,

$$\begin{aligned} P_{ij} &\stackrel{\Delta}{=} p(\mathcal{M}_k^j | \mathcal{M}_{k-1}^i, Z_{k-1}) \\ &= p(\mathcal{M}_k^j | \mathcal{M}_{k-1}^i) \end{aligned} \quad (2.45)$$

The first term of the right hand side can be treated as mode-matched filtering, which is computed by:

$$\begin{aligned} \Lambda_k^j &\stackrel{\Delta}{=} p(z_k | \mathcal{M}_k^j, Z_{k-1}) \\ &= p(z_k | \mathcal{M}_k^j, \hat{o}_{k-1}, \Sigma_{o_{k-1}}) \end{aligned} \quad (2.46)$$

To summarize, the recursive formula for computing the model probability is:

$$\mu_k^j = \eta \Lambda_k^j \sum_{i=1}^r P_{ij} \mu_{k-1}^i \quad (2.47)$$

where  $\eta$  is the normalization constant.

The *prediction* stage of Equation 2.43 can be done as follows:

$$\begin{aligned} &p(o_k | \mathcal{M}_k^j, Z_{k-1}) \\ &\stackrel{Total Prob.}{=} \sum_{i=1}^r p(o_k | \mathcal{M}_k^j, \mathcal{M}_{k-1}^i, Z_{k-1}) p(\mathcal{M}_{k-1}^i | \mathcal{M}_k^j, Z_{k-1}) \\ &\stackrel{Total Prob.}{\approx} \sum_{i=1}^r \int p(o_k | \mathcal{M}_k^j, \mathcal{M}_{k-1}^i, \{o_{k-1}^l\}_{l=1}^r) d o_{k-1} \mu^{i|j} \\ &\stackrel{Interaction}{\approx} \sum_{i=1}^r \int p(o_k | \mathcal{M}_k^j, \mathcal{M}_{k-1}^i, \hat{o}_{k-1}^i) d \hat{o}_{k-1} \mu^{i|j} \end{aligned} \quad (2.48)$$

The second line of the above equation shows that  $Z_{k-1}$  is summarized by  $r$  model-conditioned estimates and covariances, which is used in the GPB2 algorithm. The third line shows the key idea of the IMM algorithm which uses a mixing estimate  $\hat{o}_{k-1}$  as the

input of the filter instead of  $\{o_{k-1}^l\}_{l=1}^r$ . The last term on the right hand side, the mixing probability can be obtained by:

$$\begin{aligned}\mu^{i|j} &\triangleq p(\mathcal{M}_{k-1}^i | \mathcal{M}_k^j, Z_{k-1}) \\ &= \eta p(\mathcal{M}_k^j | \mathcal{M}_{k-1}^i, Z_{k-1}) p(\mathcal{M}_{k-1}^i | Z_{k-1}) \\ &= \eta P_{ij} \mu_{k-1}^i\end{aligned}\quad (2.49)$$

where  $\eta$  is the normalization constant. Using the assumption that the mixture estimate is a Gaussian and then approximating this mixture via moment matching by a single Gaussian, the mixed initial condition can be computed by:

$$o_{k-1}^{0j} = \sum_{i=1}^r \hat{o}_{k-1}^i \mu^{i|j} \quad (2.50)$$

and the corresponding covariance is:

$$\Sigma_{o_{k-1}^{0j}} = \sum_{i=1}^r \{\Sigma_{o_{k-1}^i} + (\hat{o}_{k-1}^{0j} - \hat{o}_{k-1}^i)(\hat{o}_{k-1}^{0j} - \hat{o}_{k-1}^i)\} \mu^{i|j} \quad (2.51)$$

With the mixed initial conditions, the prediction and update stages can be done with each model using Kalman filtering. Let the estimate and the corresponding covariance from each model be denoted by  $\hat{o}_k^j$  and  $\Sigma_{o_k}^j$  respectively. For output purposes, the state estimate and covariance can be combined according to the mixture equations:

$$\hat{o}_k = \sum_{j=1}^r \hat{o}_k^j \mu_k^j \quad (2.52)$$

$$\Sigma_{o_k} = \sum_{j=1}^r \{\Sigma_{o_k}^j + (\hat{o}_k^j - \hat{o}_k)(\hat{o}_k^j - \hat{o}_k)^T\} \mu_k^j \quad (2.53)$$

### Calculation Procedures of the IMM algorithm

From the Bayesian formula of the moving object tracking problem, one cycle of the the calculation procedures consists of the initialization, prediction, data association and update stages.

**Stage 1: Initialization.** Figure 2.18 shows the initialization stage of moving object tracking and Figure 2.19 shows the corresponding DBN. In this stage, it is assumed that there are  $r$  possible models in the model set, and the prior model probabilities and the mode transition probabilities are given. The mixing probabilities are computed by Equation 2.49 and the mixed initial conditions are computed by Equation 2.50 and Equation 2.51.

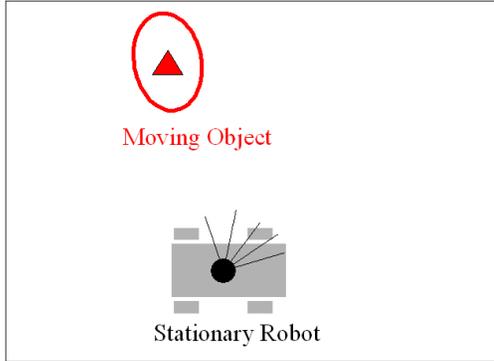


Figure 2.18. The initialization stage of moving object tracking.

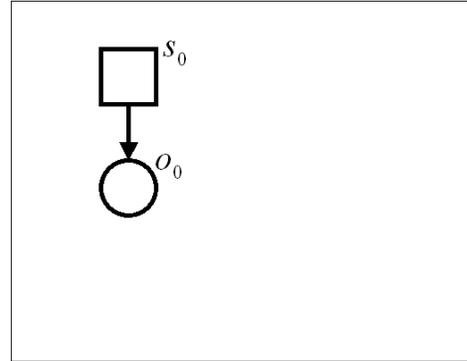


Figure 2.19. A DBN representing the initialization stage of moving object tracking.

**Stage 2: Prediction.** Figure 2.20 shows the prediction stage of moving object tracking and Figure 2.21 shows the corresponding DBN. With the mixed initial conditions, each filter use its corresponding motion model to perform prediction individually in the IMM algorithm.

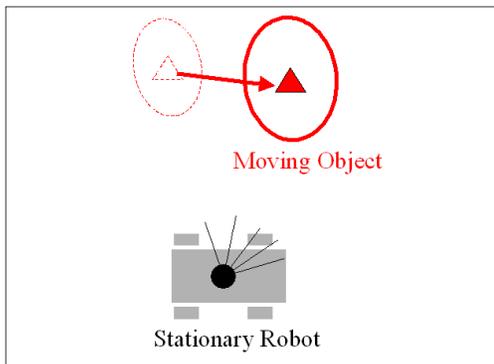


Figure 2.20. The prediction stage of moving object tracking

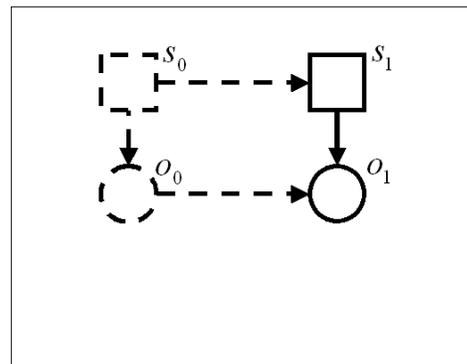


Figure 2.21. A DBN representing the prediction stage of moving object tracking

**Stage 3: Data Association.** Figure 2.22 shows the data association stage of moving object tracking and Figure 2.21 shows the corresponding DBN. In this stage, the sensor returns a new measurement  $z_k$  and each filter use its own prediction to perform data association.

**Stage 4: Update.** Figure 2.24 shows the update stage of moving object tracking and Figure 2.25 shows the corresponding DBN. In this stage, each filter is updated with the associated measurement and then the mode-matched filtering is done by Equation 2.46.

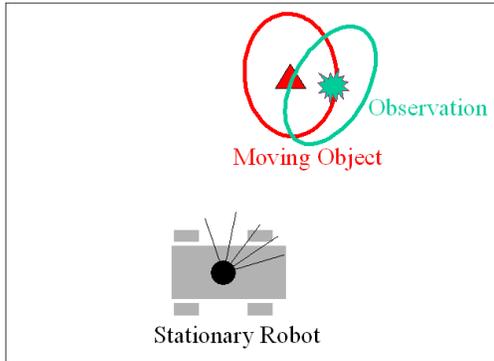


Figure 2.22. The data association stage of moving object tracking

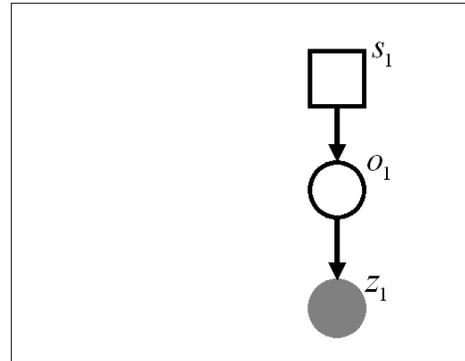


Figure 2.23. A DBN representing the data association stage of moving object tracking

The model probabilities are updated by Equation 2.47. For output purposes, the state and covariance can be computed by Equation 2.52 and Equation 2.53.

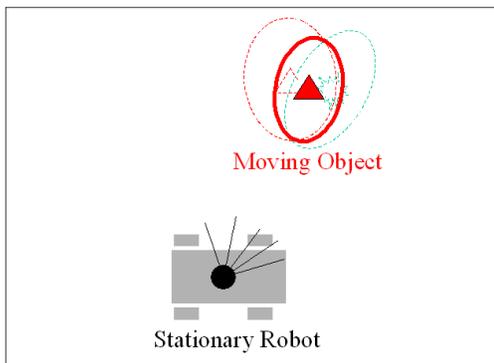


Figure 2.24. The update stage of moving object tracking

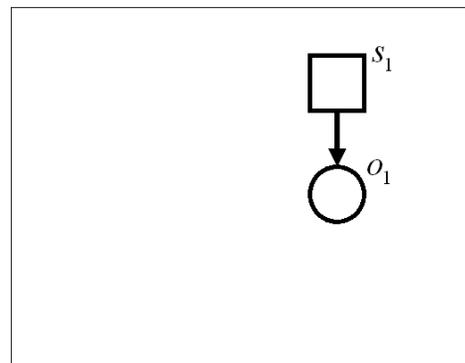


Figure 2.25. A DBN representing the update stage of moving object tracking

### Motion Modelling

In the described formulation of moving object tracking, it is assumed that a model set is given or selected in advance, and tracking is performed based on model averaging of this model set. Theoretically and practically, the performance of moving object tracking strongly relates to the selected motion models. Figure 2.26 illustrates the different performances using different motion models. Given the same data set, the tracking results differ according to the selected motion models. Figure 2.27 illustrates the effects of model set completeness. If a model set does not contain a stationary motion model, move-stop-move object tracking may not be performed well. We will address the motion modelling related issues in Chapter 4.

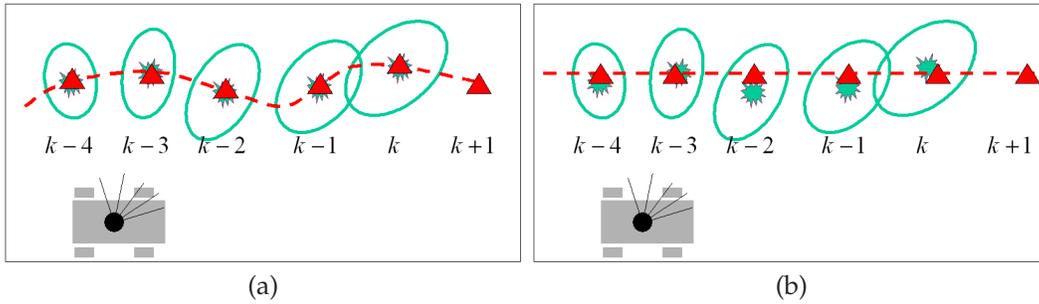


Figure 2.26. Model Selection. On the left is the result of tracking using a complicated motion model. On the right is the same data using a simple motion model.

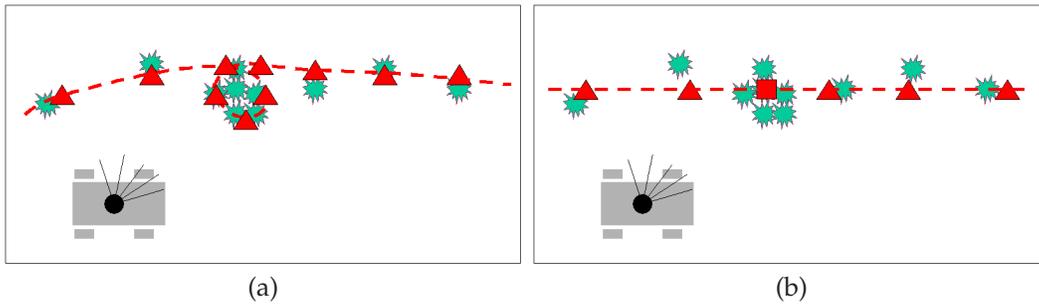


Figure 2.27. Move-stop-move object tracking. On the left is the result of tracking using only moving motion models. On the right is the result of tracking using moving motion models and a stationary motion model.

### Perception Modelling and Data Association

Regarding perception modelling, it is assumed that objects can be represented by point-features in the described formulation. In practice this may not be appropriate because of a wide variety of moving objects in urban and suburban areas. In Chapter 3, the hierarchical object based representation for moving object tracking will be described in detail. With regard to data association, using not only kinematic information from motion modelling but also geometric information from perception modelling will be addressed in Chapter 5.

### 2.4. SLAM with Generic Objects

In this section, we will present the first approach to the simultaneous localization, mapping and moving object tracking problem, SLAM with generic objects. Without making any hard decisions about whether an object is stationary or moving, the whole problem can be handled by calculating a joint posterior over all objects (robot pose, stationary objects, moving objects). Such an approach would be similar to existing SLAM algorithms, but with additional structure to allow for motion learning of the moving objects.

The formalization of SLAM with generic objects is straightforward. Similar to the moving object tracking problem, first we define that the generic object is a hybrid state consisting of the state and the motion mode.

$$\mathbf{y}_k^i \triangleq \{y_k^i, s_k^i\} \quad \text{and} \quad \mathbf{Y}_k \triangleq \{\mathbf{y}_k^1, \mathbf{y}_k^2, \dots, \mathbf{y}_k^l\} \quad (2.54)$$

where  $l$  is the number of generic objects. We then use this hybrid variable  $\mathbf{Y}$  to replace the variable  $M$  in Equation 2.15 and the Bayesian formula of SLAM with generic objects is given as:

$$\underbrace{p(x_k, \mathbf{Y}_k \mid Z_k, U_k)}_{\text{Posterior at } k} \propto \underbrace{p(z_k \mid x_k, \mathbf{Y}_k)}_{\text{Update}} \cdot \underbrace{\int \int p(x_k \mid x_{k-1}, u_k) p(\mathbf{Y}_k \mid \mathbf{Y}_{k-1}) \underbrace{p(x_{k-1}, \mathbf{Y}_{k-1} \mid Z_{k-1}, U_{k-1})}_{\text{Posterior at } k-1} dx_{k-1} d\mathbf{Y}_{k-1}}_{\text{Prediction}} \quad (2.55)$$

Figure 2.28 shows a DBN representing the SLAM with generic objects of duration three with two generic objects, which integrates the DBNs of the SLAM problem and the MOT problem.

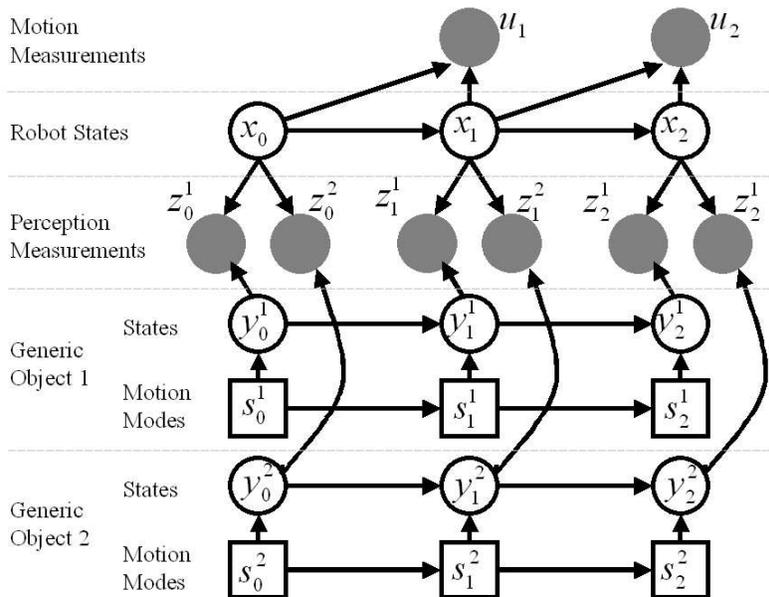


Figure 2.28. A DBN for SLAM with Generic Objects. It is an integration of the DBN of the SLAM problem (Figure 2.5) and the DBN of the MOT problem (Figure 2.14).

For SLAM with generic objects, motion modelling of generic objects is critical. A general mechanism solving motion modelling of stationary objects, moving objects and objects between stationary and moving has to be developed. We will describe a move-stop hypotheses tracking algorithm in Chapter 4 for dealing with this issue.

In the framework of SLAM with generic objects, the robot, stationary objects and moving objects are generally correlated through the convolution process in the update stage. Although the formulation of SLAM with generic objects is elegant, it is clear that SLAM with generic objects is more computationally demanding than SLAM due to the required motion modelling of all generic objects at all time steps.

The framework of SLAM with generic objects indicates that measurements belonging to *moving* objects contribute to localization and mapping as well as measurements belonging to stationary objects. Nevertheless, highly maneuverable objects are difficult to track and often unpredictable in practice. Including them in localization and mapping would have a negative effect on the robot's localization.

In next section, we will present the second approach to the whole problem, SLAM with Detection and Tracking of Moving Objects.

## 2.5. SLAM with Detection and Tracking of Moving Objects

Because of the computational intractability of SLAM with generic objects, SLAM with Detection and Tracking of Moving Objects, or SLAM with DATMO, decomposes the estimation problem into two separate estimators. Although the derivation of SLAM with DATMO is not as simple as SLAM with generic objects, the computation of SLAM with DATMO is considerably simplified in which it is possible to update both the SLAM filter and the DATMO filter in real-time. The resulting estimation problems are much lower dimensional than the joint estimation problem by maintaining separate posteriors for the stationary objects and the moving objects. In this section, we address the formulation and calculation procedures of SLAM with DATMO.

### Formulation of SLAM with DATMO

The derivation described in this section is identical to (Wang et al., 2003b).

**Assumptions.** Before introducing the derivation, the assumptions of SLAM with DATMO are addressed. The first assumption is that measurements can be decomposed into measurements of static and moving objects:

$$z_k = z_k^o + z_k^m \quad \text{and hence} \quad Z_k = Z_k^o + Z_k^m \quad (2.56)$$

Here the sensor measurement belonging to stationary objects is denoted by the variable  $z_k^m$  and the sensor measurement belonging to moving objects is denoted by the variable  $z_k^o$ . In particular this implies the following conditional independence

$$\begin{aligned} p(z_k | O_k, M, x_k) &= p(z_k^o | O_k, M, x_k) p(z_k^m | O_k, M, x_k) \\ &= p(z_k^o | O_k, x_k) p(z_k^m | M, x_k) \end{aligned} \quad (2.57)$$

where the variable  $x_k$  denotes the true pose of the robot at time  $k$ , and the variable  $M = \{m_k^1, m_k^2, \dots, m_k^l\}$  denotes the true locations of the stationary objects, of which there are  $l$  in the world at time  $k$ . The variable  $O_k = \{o_k^1, o_k^2, \dots, o_k^n\}$  denotes the true states of the moving objects, of which there are  $n$  in the world at time  $k$ .

The second assumption is that when estimating the posterior over the map and the robot pose, the measurements of moving objects carry no information, neither do their positions  $O_k$ :

$$p(M, x_k | O_k, Z_k, U_k) = p(M, x_k | Z_k^m, U_k) \quad (2.58)$$

where the variable  $U_k = \{u_1, u_2, \dots, u_k\}$  denotes the motion measurements up to time  $k$ .

This is correct if we have no information whatsoever about the speed at which objects move. Here it is an approximation, but one that reduces the complexity of SLAM with moving features enormously.

**Derivation.** We begin by factoring out the most recent measurement:

$$p(O_k, M, x_k | Z_k, U_k) \propto p(z_k | O_k, M, x_k, Z_{k-1}, U_k) p(O_k, M, x_k | Z_{k-1}, U_k) \quad (2.59)$$

Observing the standard Markov assumption, we note that  $p(z_k | O_k, M, x_k, Z_{k-1}, U_k)$  does not depend on  $Z_{k-1}, U_k$ , hence we have

$$p(O_k, M, x_k | Z_k, U_k) \propto p(z_k | O_k, M, x_k) p(O_k, M, x_k | Z_{k-1}, U_k) \quad (2.60)$$

Furthermore, we can now partition the measurement  $z_k = z_k^o + z_k^m$  into moving and static, and obtain by exploiting the first assumption and Equation 2.57:

$$p(O_k, M, x_k | Z_k, U_k) \propto p(z_k^o | O_k, x_k) p(z_k^m | M, x_k) p(O_k, M, x_k | Z_{k-1}, U_k) \quad (2.61)$$

The rightmost term  $p(O_k, M, x_k | Z_{k-1}, U_k)$  can now be further developed, exploiting the second assumption

$$\begin{aligned} p(O_k, M, x_k | Z_{k-1}, U_k) &= p(O_k | Z_{k-1}, U_k) p(M, x_k | O_k, Z_{k-1}, U_k) \\ &= p(O_k | Z_{k-1}, U_k) p(M, x_k | Z_{k-1}^m, U_k) \end{aligned} \quad (2.62)$$

Hence we get for our desired posterior

$$\begin{aligned}
& p(O_k, M, x_k \mid Z_k, U_k) \\
& \propto p(z_k^o \mid O_k, x_k) p(z_k^m \mid M, x_k) p(O_k \mid Z_{k-1}, U_k) p(M, x_k \mid Z_{k-1}^m, U_k) \\
& \propto \underbrace{p(z_k^o \mid O_k, x_k) p(O_k \mid Z_{k-1}, U_k)}_{\text{DATMO}} \underbrace{p(z_k^m \mid M, x_k) p(M, x_k \mid Z_{k-1}^m, U_k)}_{\text{SLAM}} \quad (2.63)
\end{aligned}$$

The term  $p(O_k \mid Z_{k-1}, U_k)$  resolves to the following prediction

$$\begin{aligned}
p(O_k \mid Z_{k-1}, U_k) &= \int p(O_k \mid Z_{k-1}, U_k, O_{k-1}) p(O_{k-1} \mid Z_{k-1}, U_k) dO_{k-1} \\
&= \int p(O_k \mid O_{k-1}) p(O_{k-1} \mid Z_{k-1}, U_{k-1}) dO_{k-1} \quad (2.64)
\end{aligned}$$

Finally, the term  $p(M, x_k \mid Z_{k-1}^m, U_k)$  in Equation 2.63 is obtained by the following step:

$$\begin{aligned}
& p(M, x_k \mid Z_{k-1}^m, U_k) \\
&= p(x_k \mid Z_{k-1}^m, U_k, M) p(M \mid Z_{k-1}^m, U_k) \\
&= \int p(x_k \mid Z_{k-1}^m, U_k, M, x_{k-1}) p(x_{k-1} \mid Z_{k-1}^m, U_k, M) p(M \mid Z_{k-1}^m, U_k) dx_{k-1} \\
&= \int p(x_k \mid u_k, x_{k-1}) p(x_{k-1}, M \mid Z_{k-1}^m, U_{k-1}) dx_{k-1} \quad (2.65)
\end{aligned}$$

which is the familiar SLAM prediction step. Putting everything back into Equation 2.63 we now obtain the final filter equation:

$$\begin{aligned}
& p(O_k, M, x_k \mid Z_k, U_k) \\
& \propto \underbrace{p(z_k^o \mid O_k, x_k)}_{\text{Update}} \underbrace{\int p(O_k \mid O_{k-1}) p(O_{k-1} \mid Z_{k-1}, U_{k-1}) dO_{k-1}}_{\text{Prediction}} \\
& \quad \underbrace{p(z_k^m \mid M, x_k)}_{\text{Update}} \underbrace{\int p(x_k \mid u_k, x_{k-1}) p(x_{k-1}, M \mid Z_{k-1}^m, U_{k-1}) dx_{k-1}}_{\text{Prediction}} \quad (2.66)
\end{aligned}$$

**Solving the SLAM with DATMO problem.** From Equation 2.66, input to this SLAM with DATMO filter are two separate posteriors, one of the conventional SLAM form,  $p(x_{k-1}, M \mid Z_{k-1}^m, U_{k-1})$ , and a separate one for DATMO,  $p(O_{k-1} \mid Z_{k-1}, U_{k-1})$ .

The remaining question is now how to recover those posteriors at time  $k$ . For the SLAM part, the recovery is simple:

$$\begin{aligned}
& p(x_k, M \mid Z_k^m, U_k) \\
&= \int p(O_k, M, x_k \mid Z_k, U_k) dO_k \\
&\propto p(z_k^m \mid M, x_k) \int p(x_k \mid u_k, x_{k-1}) p(x_{k-1}, M \mid Z_{k-1}^m, U_{k-1}) dx_{k-1} \quad (2.67)
\end{aligned}$$

For DATMO, we get

$$\begin{aligned}
& p(O_k | Z_k, U_k) \\
&= \int \int p(O_k, M, x_k | Z_k, U_k) dM dx_k \\
&\propto \int \left[ p(z_k^o | O_k, x_k) \int p(O_k | O_{k-1}) p(O_{k-1} | Z_{k-1}, U_{k-1}) dO_{k-1} \right] \\
&\quad p(x_k | Z_k^m, U_k) dx_k \tag{2.68}
\end{aligned}$$

where the posterior over the pose  $p(x_k | Z_k^m, U_k)$  is simply the marginal of the joint calculated in Equation 2.67:

$$p(x_k | Z_k^m, U_k) = \int p(x_k, M | Z_k^m, U_k) dM \tag{2.69}$$

For Gaussians, all these integrations are easily carried out in closed form. Equation 2.68 shows that DATMO should take account of the uncertainty in the pose estimate of the robot because the perception measurements are directly from the robot.

### Calculation Procedures

Figure 2.29 shows a DBN representing three time steps of an example SLAM with DATMO problem with one moving object and one stationary object. The calculation procedures for solving the SLAM with DATMO problem are the same as the SLAM problem and the moving object tracking problem, which consist of the initialization, prediction, data association and update stages. In this section, we explain the procedures to compute Equation 2.66 with the visualization aid from DBN.

**Stage 1: Initialization.** Figure 2.30 shows the initialization stage. In this example, two stationary objects and one moving object are initialized. Figure 2.31 shows a DBN representing this example. It is assumed that the measurements can be classified into the measurements of stationary objects and moving objects.

**Stage 2: Prediction.** Figure 2.32 shows the prediction stage in which the robot gets a new motion measurement. Only the robot and the moving object are predicted. The robot motion prediction is done with the robot motion model and the new motion measurement. However, there is no motion measurement associated with the moving object, and the motion model of this moving object is unknown. In this dissertation, the IMM algorithm is applied. The moving object motion prediction is done with the mixed initial conditions from the selected motion models. Figure 2.33 shows a DBN representing the prediction stage.

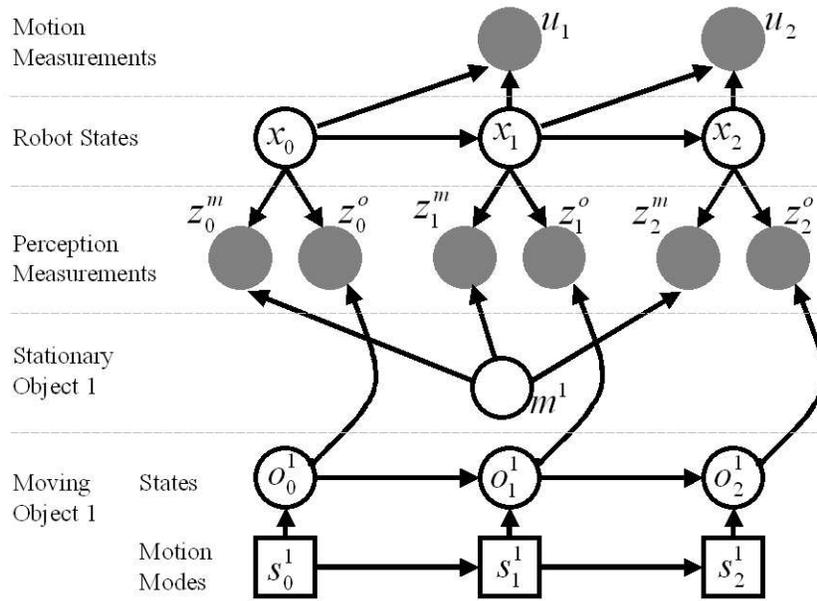


Figure 2.29. A DBN of the SLAM with DATMO problem of duration three with one moving object and one stationary object.

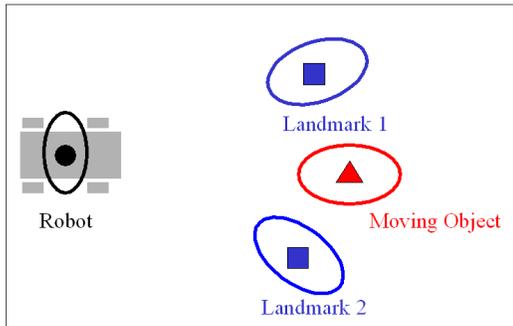


Figure 2.30. The initialization stage of SLAM with DATMO

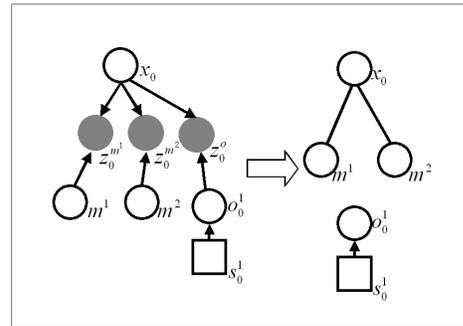


Figure 2.31. A DBN representing the initialization stage of SLAM with DATMO

**Stage 3: Data Association.** Figure 2.34 shows the data association stage in which the robot gets a new perception measurement at the new location. The new perception measurement is associated with the stationary objects and the moving object. Figure 2.35 shows a DBN representing this stage.

**Stage 4: Update of the SLAM part.** Figure 2.36 shows the update stage of the SLAM part of the whole problem. Only measurements associated with stationary objects are used to update the robot pose and the stationary object map. After this update, the map and the robot pose are more accurate. Figure 2.37 shows a DBN representing this stage.

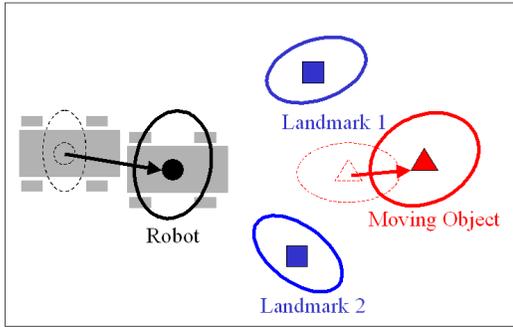


Figure 2.32. The prediction stage of SLAM with DATMO

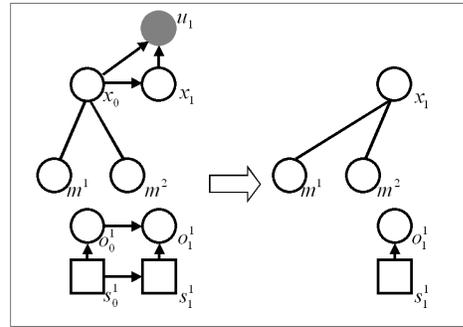


Figure 2.33. A DBN representing the prediction stage of SLAM with DATMO

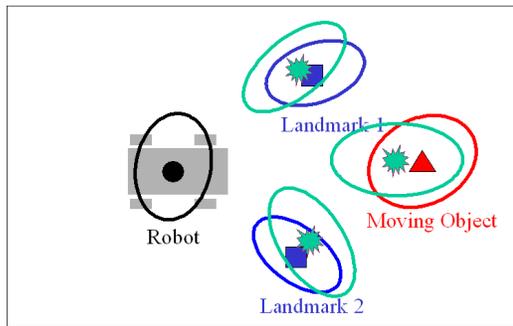


Figure 2.34. The data association stage of SLAM with DATMO

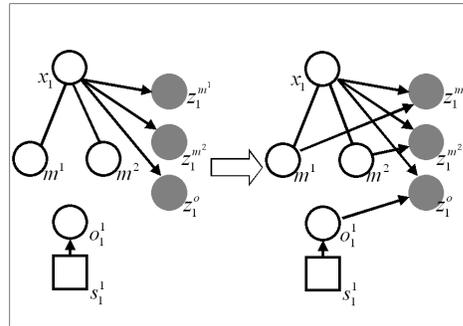


Figure 2.35. A DBN representing the data association stage of SLAM with DATMO

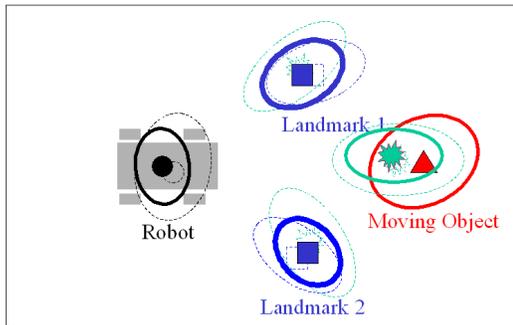


Figure 2.36. The update stage of the SLAM part of SLAM with DATMO

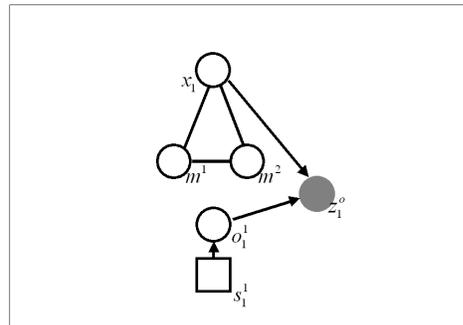


Figure 2.37. A DBN representing the update stage of the SLAM part of SLAM with DATMO

**Stage 5: Update of the DATMO part.** Figure 2.38 shows the update stage of the DATMO part of the whole problem. Because the robot pose estimate is more accurate after the update of the SLAM part, the measurement associated with the moving object is more

accurate as well. In the update stage of the DATMO part, this more accurate measurement is used to update the moving object pose as well as its motion models.

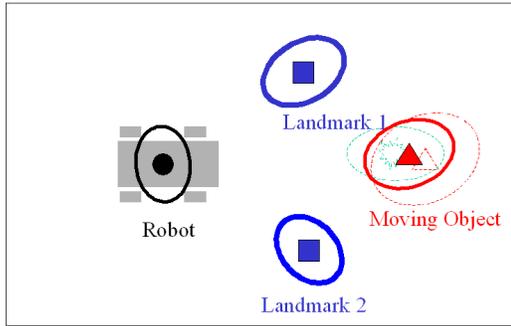


Figure 2.38. The update stage of the DATMO part of SLAM with DATMO

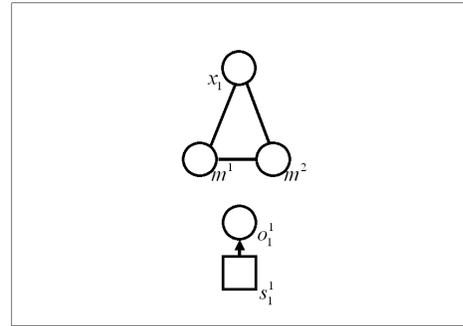


Figure 2.39. A DBN representing the update stage of the DATMO part of SLAM with DATMO

## 2.6. Summary

In this chapter, we have established the foundations of the simultaneous localization, mapping and moving object tracking problem. We described the formulas for manipulating uncertain spatial relationships, and reviewed the probabilistic formulas of the SLAM problem and the moving object tracking problem. We treated simultaneous localization, mapping and moving object tracking as a new discipline at the intersection of SLAM and moving object tracking, and described two solutions, SLAM with generic objects and SLAM with DATMO. The corresponding formulas provide a solid basis for understanding and solving the whole problem.

In addition to the established foundations, we need to eliminate the gaps between the foundations and implementation for solving the whole problem from ground vehicles at high speeds in urban areas. These gaps arise from a number of implicit assumptions in terms of perception modelling, motion modelling and data association, which will be addressed in the later chapters. In the next chapter, we will discuss the issues of perception modelling.

## CHAPTER 3

---

### Perception Modelling

*A photograph is a secret about a secret. The more it tells you, the less you know.*  
– Diane Arbus

**P**ERCEPTION MODELLING, or *representation*, provides a bridge between *perception sensor measurements* and *theory*; different representation methods lead to different means to calculate the theoretical formulas. Representation should allow information from different sensors, from different locations and from different time frames to be fused.

In the tracking literature, targets are usually represented by *point-features*. In most air and sea vehicle tracking applications, the geometrical information of the targets is not included because of the limited resolution of perception sensors. However, the signal-related data such as the amplitude of the signal can be included to aid data association and classification. On the other hand, research on mobile robot navigation has produced four major paradigms for environment representation: feature-based approaches (Leonard and Durrant-Whyte, 1991), grid-based approaches (Elfes, 1988; Thrun et al., 1998), direct approaches (Lu and Milios, 1994, 1997), and topological approaches (Choset and Nagatani, 2001). Because topological maps are usually generated on top of grid-based or feature-based maps by partitioning grid-based or feature-based maps into coherent regions, we will only focus on feature-based approaches, grid-based approaches and direct approaches.

First, these three paradigms will be compared in terms of:

- *Uncertainty management,*
- *Sensor characteristics,*
- *Environment representability,*
- *Data compression,*
- *Loop-closing mechanism.*

All of these issues should be considered in both the moving object tracking problem and the SLAM problem except the last term, loop-closing mechanism.

The comparison will show that these paradigms are problematic and not sufficient for large, outdoor environments. In order to overcome the difficulties, a *hierarchical object* based approach to hierarchically integrate the direct method, the grid-based approach and the feature-based method is presented.

### 3.1. Perception Models

In this section, we will discuss the advantages and disadvantages of the feature-based, grid-based and direct approaches.

#### Feature-based methods

Feature (landmark) based approaches compress raw data into predefined features. They provide an elegant way such as the EKF-based approaches to manage uncertainty of localization and mapping. The loop closing mechanism is seamlessly embedded by maintaining the covariance matrix given correct data association.

For most indoor applications, lines, circles, corners and other simple geometrical features are rich and easy to detect. (Pfister et al., 2003) present a weighted matching algorithm to take *sensor characteristics* as well as *correspondence error* into account. But their method cannot be expended to unstructured outdoor environments because of the planar environment assumption.

For outdoor applications, extracting features robustly and correctly is extremely difficult because outdoor environments contain many different kinds of objects. For example, bushes (see Figure 3.1), trees, or curvy objects (see Figure 3.2) have shapes which are hard to define. In these kinds of environments, whenever a feature is extracted an error from feature extraction will be produced because of wrong predefined features. Figure 3.3 illustrates that the results of circle extraction of the same object using measurements from different positions are different. Figure 3.4 illustrates the ambiguity of line extraction for a curved object.

#### Grid-based methods

Grid-based methods use a cellular representation called *Occupancy Grids* (Matthies and Elfes, 1988) or *Evidence Grids* (Martin and Moravec, 1996). Mapping is accomplished using a Bayesian scheme, and localization can be accomplished using correlation of a sensor scan with the grid map (Konolige and Chou, 1999).

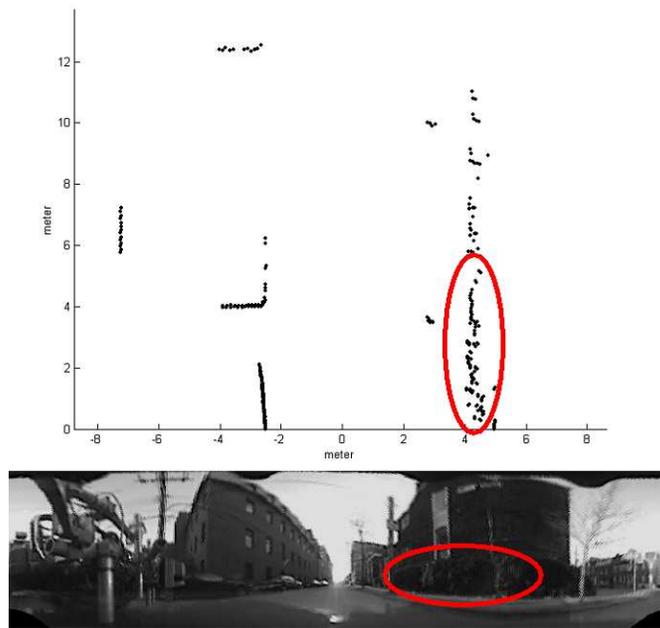


Figure 3.1. Vegetation and plant object: Bush. The ellipses indicate the bush area in the SICK scan data and in the unwarped image from the omni-directional camera.

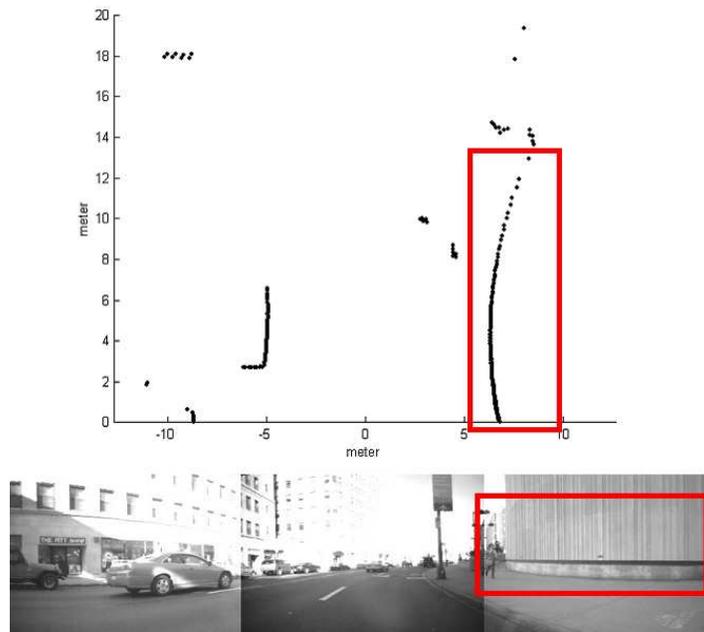


Figure 3.2. Curvy object: A building. The rectangles indicate a curvy building in the SICK scan data and in the image from the three-camera system.

In terms of *sensor characteristics* and *environment representability*, grid-based approaches are more advanced than feature-based approaches. Grid-maps can represent any kinds of environments and the quality of the map can be adjusted by adapting the grid resolution.

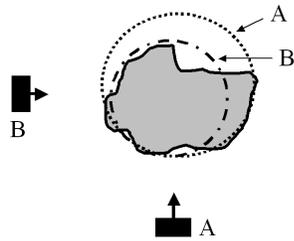


Figure 3.3. Circle extraction.

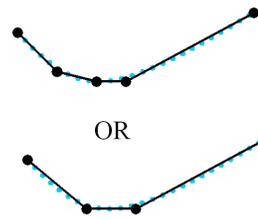


Figure 3.4. Line extraction.

Grid-based approaches are specially suitable for noisy sensors such as stereo camera, sonar and radar in which features are hard to define and extract from highly uncertain and uninformative measurements.

Nevertheless, grid-based approaches do not provide a mechanism for *loop closing*. Recall that correlation between the robot and landmarks is explicitly managed by the covariance matrix or the information matrix in the feature-based approaches. Correlation between the robot and landmarks is implicitly embedded in Occupancy Grids. How to retrieve correlation from Occupancy Grids is an open question. Given that a loop is correctly detected, loop closing can not be done with the existing grids. Additional computation power is needed to run consistent pose estimation algorithms such as (Lu and Milios, 1997; Kelly and Unnikrishnan, 2003) and the previous raw scans have to be used to generate a new global consistent map (Gutmann and Konolige, 1999; Thrun et al., 2000).

### Direct methods

Direct methods represent the physical environment using raw data points without extracting predefined features.

Localization can be done by using range image registration algorithms from the computer vision literature. For instance, the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992; Chen and Medioni, 1991; Zhang, 1994) is a widely used direct method; many variants have been proposed based on the basic ICP concept (Rusinkiewicz and Levoy, 2001). However, a good initial prediction of the transformation between scans is required because of its heuristic assumption for data association. When transformation between scans is unavailable, a number of technique can be used to recognize scans and provide a good initial prediction of the transformation such as *Spin Image* (Johnson, 1997), principle moments (Besl and McKay, 1992), normal of distinctive points (Potmesil, 1983), and principle curvatures (Feldmar and Ayache, 1994).

The map is represented as a *list* of raw scans. Because there is overlap between scans, memory requirement for storing the map can be reduced by the integration (merging) process such as (Garland and Heckbert, 1998) and the map can be represented as *triangular meshes*<sup>1</sup> (Turk and Levoy, 1994), *parametric surfaces* (Menq et al., 1992), and *octrees* (Champleboux et al., 1992). Just as with the grid-based approaches, when loops are detected additional computation power is needed to run consistent pose estimation algorithms and the previous raw scans are used to generate a global consistent map.

In terms of *uncertainty management* and *sensor characteristics*, very little work addresses how to quantify the uncertainty of the transformation estimate from registration process. Uncertainty arises mainly from outliers, wrong correspondences, and measurement noises. Without taking measurement noise into account, several methods to estimate the covariance matrix of the pose estimate were proposed by (Gutmann and Schlegel, 1996), (Simon, 1996), (Lee et al., 2002), and (Bengtsson and Baerveldt, 2001). Compared to indoor applications, the distances between objects and sensors in outdoor environments are usually much longer, which make measurements more uncertain and sparse. By assuming measurement noise is Gaussian, (Pennec and Thirion, 1997) used the extended Kalman filter to estimate both the rigid transformation and its covariance matrix. However, this approach is very sensitive to correspondence errors. Additionally, the assumption that the uncertainty of the pose estimate from registration processes can be modelled by Gaussian distributions is not always valid.

### Comparison

To summarize, we show the comparison of different representations in Table 3.1. With regard to uncertainty management and loop closing mechanism, feature-based approaches have an elegant means. Regarding sensor characteristics, grid-based approaches are the easiest to implement and the most suitable for imprecise sensors such as sonar and radar. Respecting environment representability, feature-based approaches are limited to indoor or structured environments in which features are easy to define and extract.

### 3.2. Hierarchical Object based Representation

Because none of these three main paradigms is sufficient for large, outdoor environments, we propose a hierarchical object based representation to integrate these paradigms and to overcome their disadvantages.

---

<sup>1</sup>The web page, <http://www.cs.cmu.edu/~ph/>, provides comprehensive link collections.

**Table 3.1.** An attempt to compare the different representation methods. ✓ indicates that the method is elegant and appropriate. △ indicates that extra work is needed or the method is inapplicable.

Representations	Feature-based	Grid-based	Direct
Uncertainty management	✓	✓	△
Loop closing mechanism	✓	△	△
Sensor characteristics	△	✓	✓
Environment representability	△	✓	✓
Data Compression	✓	△	△

In outdoor or urban environments, features are extremely difficult to define and extract because both stationary and moving objects do not have specific sizes and shapes. Therefore, instead of using an ad hoc approach to define features in specific environments or for specific objects, *free-form objects* are used.

At the preprocessing stage, scans (perception measurements) are grouped into *segments* using a simple distance criterion. The segments over different time frames are integrated into *objects* after localization, mapping and tracking processes. Instead of using *track* in tracking terminology, *segment* is used because of the perception sensor used in this work. Because not only moving targets but also stationary landmarks are tracked in the whole process, the more general term, *object*, is used instead of the term, *target*.

Registration of scan segments over different time frames is done by using the *direct* method, namely the ICP algorithm. Because range images are sparser and more uncertain in outdoor applications than indoor applications, the pose estimation and the corresponding distribution from the ICP algorithm are not reliable. For dealing with the sparse data issues, a sampling-based approach is used to estimate the uncertainty from correspondence errors. For dealing with the uncertain data issues, a correlation-based approach is used with the *grid-based* method for estimating the uncertainty from measurement noise. For loop closing in large environments, the origins of the object coordinate system are used as features with the mechanism of the *feature-based* approaches.

Our approach is hierarchical because these three main representation paradigms are used on different levels. The direct method is used on the lowest level and the feature-based approach is used on the highest level. Objects are described by a state vector, or *object-feature* and a grid map, or *object-grids*. Object-features are used with the mechanism of the feature-based approaches for moving object tracking and for loop closing. Object-grids are used to take measurement noise into account for estimating object registration uncertainty and to integrate measurements over different time frames. Figure 3.5 shows an example of the hierarchical object based representation.

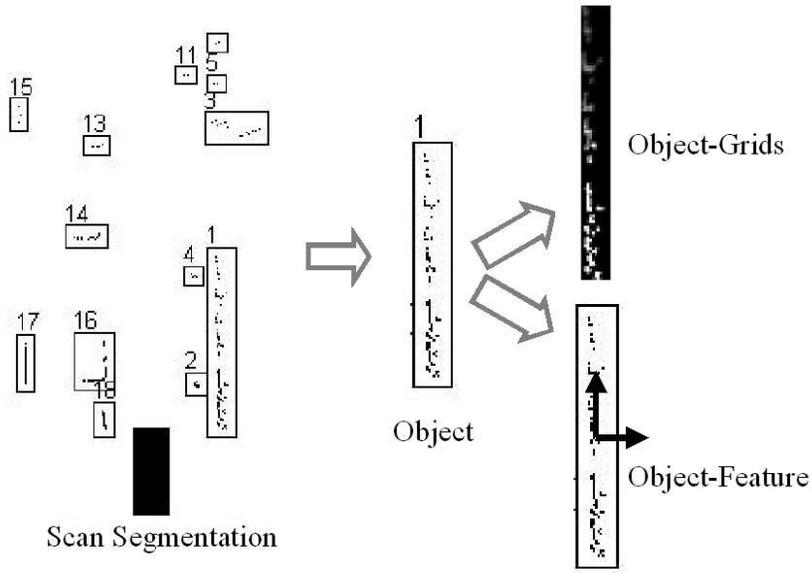


Figure 3.5. Hierarchical object based representation. The black solid box denotes the robot (2mx5m).

In this section, scan segmentation, sensor noise modelling and sparse data issues will be described. The sampling and correlation based approach for estimating the uncertainty of object registration will be addressed in Section 3.3. The hierarchical object based representation for moving object tracking and for SLAM will be addressed in Section 3.4 and Section 3.5 respectively.

### Scan Segmentation

Scan segmentation is the first stage of the hierarchical object-based approach. (Hoover et al., 1996) proposed a methodology for evaluating range image segmentation algorithms, which are mainly for segmenting a range image into planar or quadric patches. Because objects in outdoor environments do not have specific sizes and shapes, these algorithms are not suitable.

Here we use a simple distance criterion to segment measurement points into objects. Although this simple criterion can not produce perfect segmentation results, more precise segmentation will be accomplished by the localization, mapping and moving object tracking processes using spatial and temporal information over several time frames. Figure 3.6 shows an example of scan segmentation.

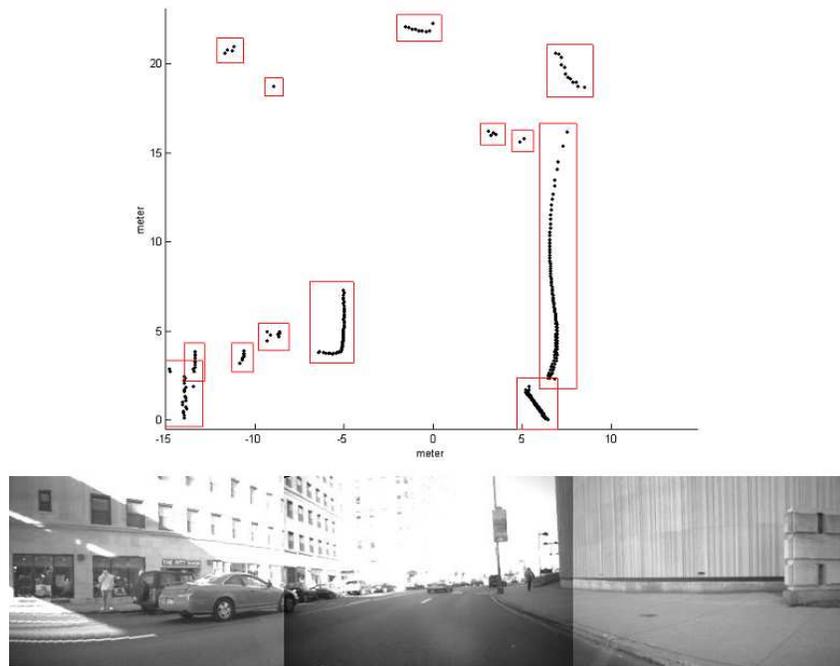


Figure 3.6. An example of scan segmentation.

### Perception Sensor Modelling

It is well known that several important physical phenomena such as the material properties of an object, the sensor incidence angle, and environmental conditions affect the accuracy of laser scanner measurements. Although laser rangefinders such as SICK laser scanners provide more accurate measurements than sonar, radar and stereo cameras, neglecting measurement noise in the localization, mapping, and moving object tracking processes may be over optimistic in situations using data collected from a platform at high speeds in outdoor environments.

According to the manual of SICK laser scanners (Sick Optics, 2003), the spot spacing of SICK LMS 211/221/291 is smaller than the spot diameter for an angular resolution of 0.5 degree. This means that footprints of consecutive measurements overlap each other. The photo in Figure 3.7 taken from an infrared camera shows this phenomenon. A red rectangle indicates a footprint of one measurement point.

With regard to range measurement error, we conservatively assume the error as 1% of the range measurement because of outdoor physical phenomena. The uncertainty of each measurement point  $z_k^i$  in the polar coordinate system is described as:

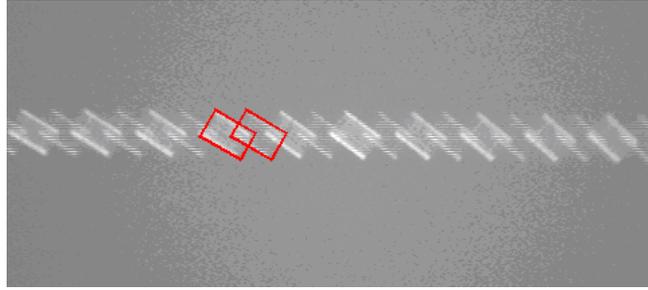


Figure 3.7. Footprints of the measurement from SICK LMS 291

$$\Sigma_{z_k^i} = \begin{bmatrix} \sigma_{r^i}^2 & 0 \\ 0 & \sigma_{\theta^i}^2 \end{bmatrix} \quad (3.1)$$

The uncertainty can be described in the Cartesian coordinate system by the head-to-tail operation described in Section 2.1. Figure 3.8 shows the SICK LMS 211/221/291 noise model.

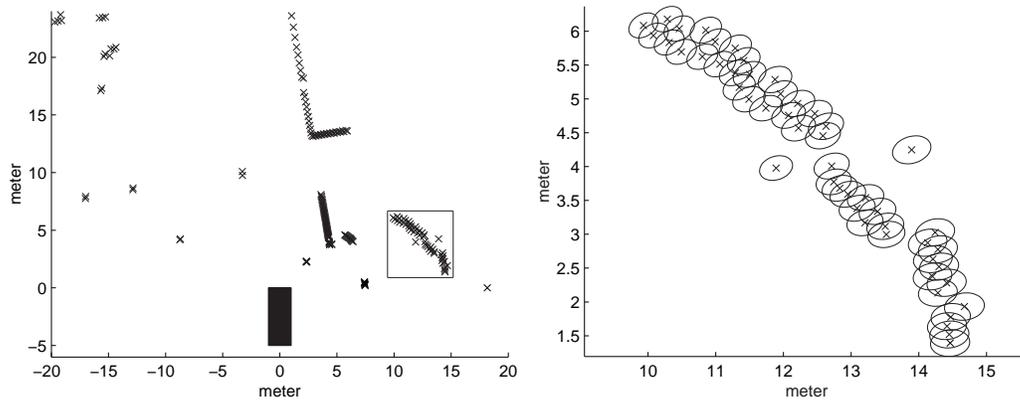


Figure 3.8. SICK LMS 211/221/291 noise model. Left: the whole scan. Right: the enlargement of the blocked region on the left. The distributions of the measurement points are shown by  $2\sigma$  ellipses (95% confidence).

In most indoor applications, it is assumed that a horizontal range scan is a collection of range measurements taken from a single robot position. When the robot is moving at high speeds, this assumption is invalid. We use the rotating rate of the scanning device and the velocity of the robot to correct the errors from this assumption.

### Sparse Data

Compared to indoor applications, the distances between objects and sensors in outdoor environments are usually much longer, which make measurements more uncertain

and not as dense. Sparse data causes problems of *data association in the small*, or *correspondence finding*, which directly affect the accuracy of direct methods. In the computer vision and indoor SLAM literature, the assumption that corresponding points present the same physical point is valid because data is dense. If a point-point metric is used in the ICP algorithm, one-to-one correspondence will not be guaranteed with sparse data, which will result in decreasing the accuracy of transformation estimation and slower convergence. Research on the ICP algorithms suggests that minimizing distances between points and tangent planes can converge faster. But because of sparse data and irregular surfaces in outdoor environments, the secondary information derived from raw data such as surface normal can be unreliable and too sensitive. A sampling-based approach for dealing with this issue will be addressed in the next section.

### 3.3. Sampling- and Correlation-based Range Image Matching

Recall that localization using a given map can be described in a probabilistic form as:

$$p(z_k | x_k, M) \quad (3.2)$$

where  $x_k$  is the predicted pose of the robot,  $M$  is the given map, and  $z_k$  is the new perception measurement. This formula can be also treated as the update stage of the SLAM problem. By replacing  $M$  and  $z_k$  with two range images  $\mathbf{A}$  and  $\mathbf{B}$  and replacing  $x_k$  with an initial guess of the relative transformation  $T'$  between these two range images, the range image registration problem can be described in the same form:

$$p(\mathbf{B} | T', \mathbf{A}) \quad (3.3)$$

When the laser scanners are used, the localization problem and the range image registration problem are identical. A number of range image registration techniques in the computer vision literature can be used for solving the localization problem.

In this section, first the ICP algorithm is introduced. Then the sampling and correlation based approach is presented for taking correspondence errors and measurement noise into account. We also define a parameter called the *object saliency score* to quantify the saliency of the object based on the covariance estimate from the sampling and correlation based range image registration algorithm.

#### The Iterated Closest Point Algorithm

Let two measurement point sets  $\mathbf{A}$  and  $\mathbf{B}$  be collected from the two true locations  $x_{wA}$  and  $x_{wB}$  in the world frame  $w$  respectively. The true relative transformation  $T$  in the  $A$

frame can be computed by:

$$\begin{aligned}\mathcal{T} &= \oplus(\ominus(x_{wA}), x_{wB}) \\ &= \oplus(x_{Aw}, x_{wB}) = x_{AB}\end{aligned}\quad (3.4)$$

where  $\oplus$  is the compounding operation and  $\ominus$  is the reverse operation defined in Chapter 2.1.

With a reasonable guess of the relative transformation  $\mathcal{T}'$ , the goal of range image registration is to find the optimal estimate  $\hat{\mathcal{T}}$  of  $\mathcal{T}$  to align these two point sets with minimal disparity. The ICP algorithm is a widely used direct method and has become the dominant method for aligning 2D and 3D range images because of its simplicity. The ICP algorithm can be summarized as follows. Using a reasonably good initial guess of the relative transformation  $\mathcal{T}'$ , a set of point is chosen from  $\mathbf{A}$ , and the corresponding closest points are found from  $\mathbf{B}$ . The better estimate of the relative transformation  $\hat{\mathcal{T}}$  is computed by using a least squares method. This procedure is iterated until the change of the estimated related transformation becomes very small. Let  $n$  corresponding point pairs be denoted by  $\{(a^i, b^i)\}_{i=1}^n$ . A distance metric can be defined as:

$$E = \sum_{i=1}^n \|\oplus(\mathcal{T}', b^i) - a^i\|^2 \quad (3.5)$$

By minimizing  $E$ , a closed-form solution can be obtained as (Lu and Milios, 1994):

$$\begin{aligned}\hat{\mathcal{T}}_\theta &= \arctan \frac{\Sigma_{b_x a_y} - \Sigma_{b_y a_x}}{\Sigma_{b_x a_x} + \Sigma_{b_y a_y}} \\ \hat{\mathcal{T}}_x &= \bar{a}_x - (\bar{b}_x \cos \hat{\mathcal{T}}_\theta - \bar{b}_y \sin \hat{\mathcal{T}}_\theta) \\ \hat{\mathcal{T}}_y &= \bar{b}_y - (\bar{a}_x \sin \hat{\mathcal{T}}_\theta + \bar{a}_y \cos \hat{\mathcal{T}}_\theta)\end{aligned}\quad (3.6)$$

where

$$\begin{aligned}\bar{a}_x &= \frac{1}{n} \sum_{i=1}^n a_x^i, \quad \bar{a}_y = \frac{1}{n} \sum_{i=1}^n a_y^i, \quad \bar{b}_x = \frac{1}{n} \sum_{i=1}^n b_x^i, \quad \bar{b}_y = \frac{1}{n} \sum_{i=1}^n b_y^i \\ \Sigma_{b_x a_x} &= \sum_{i=1}^n (b_x^i - \bar{b}_x)(a_x^i - \bar{a}_x), \quad \Sigma_{b_y a_y} = \sum_{i=1}^n (b_y^i - \bar{b}_y)(a_y^i - \bar{a}_y) \\ \Sigma_{b_x a_y} &= \sum_{i=1}^n (b_x^i - \bar{b}_x)(a_y^i - \bar{a}_y), \quad \Sigma_{b_y a_x} = \sum_{i=1}^n (b_y^i - \bar{b}_y)(a_x^i - \bar{a}_x)\end{aligned}\quad (3.7)$$

For localization, mapping and tracking, both the pose estimate and its corresponding distribution are important. In (Lu and Milios, 1997), Equation 3.5 is linearized and the analytical solution of the covariance matrix can be derived using the theory of linear regression. In (Bengtsson and Baerveldt, 2001), a Hessian matrix based method to compute

the covariance matrix was proposed. Because of the heuristic way the ICP algorithm finds corresponding points, neither method reliably estimates the uncertainty from correspondence errors. The method of Fu and Milios tends to underestimate the uncertainty and the method of Bengtsson and Baerveldt tends to overestimate the uncertainty. Additionally, these methods do not take measurement noise into account.

Because the heuristic way for finding corresponding points causes local minimum problems, a good reasonable initial guess of the relative transformation is essential for the successful usage of the ICP algorithm. Nevertheless, the *saliency* of the range images is also critical. Without a reasonable guess of the relative transformation, the ICP algorithm can still find a global minimum solution as long as the sensed scene has enough salient features or a high saliency score. The following figures illustrate the object saliency effect. Figure 3.9 shows two scans from a static environment and the scan segmentation results.

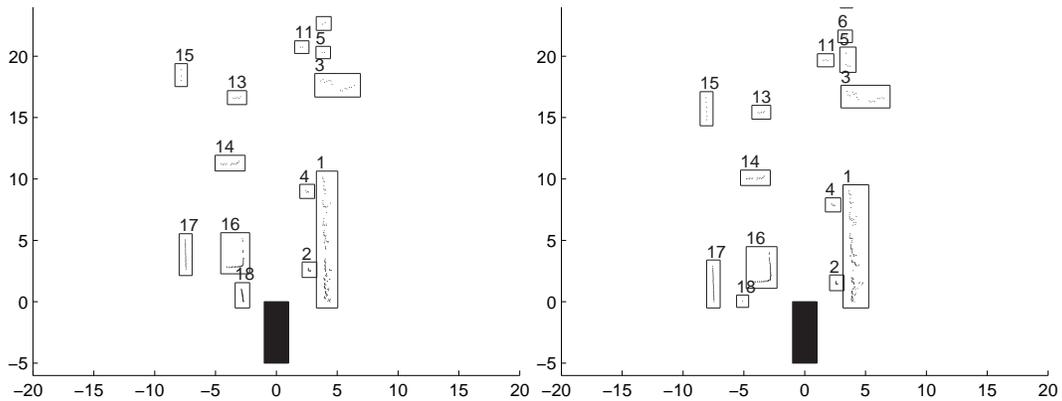


Figure 3.9. Left: Scan **A**. Right: Scan **B**. The solid box denotes the robot (2mx5m). Segmentation results are shown with segment numbers.

In this example, we assume that the motion measurement is unavailable and the initial guess of the relative transformation is zero. Figure 3.10 shows this initial guess of the relative transformation.

In order to illustrate the object saliency effect, range images **A** and **B** are aligned using the same initial relative transformation guess but using different scan segments: one is matching with only segment 1 of scan **A** and segment 1 of scan **B**; the other is matching with the whole scans of **A** and **B**. Figure 3.11 shows the registration results. It seems that the ICP algorithm provides satisfactory results in both cases and it is hard to quantify which result is better. However, by comparing the results with the whole scans in Figure 3.12, it is easy to justify that registration using only scan segment 1 of **A** and **B** provides a local minimum solution instead of the global one.

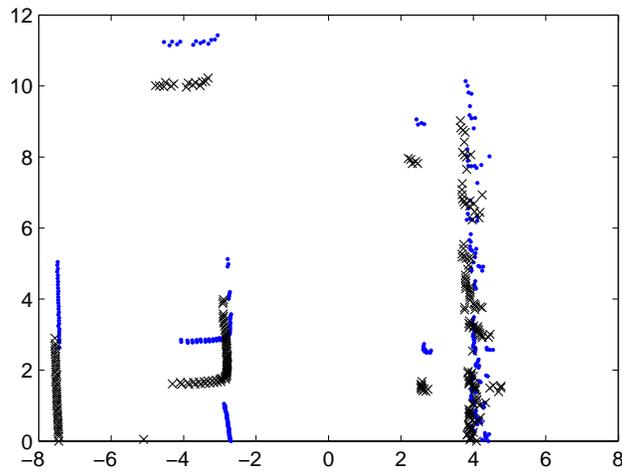


Figure 3.10. An initial guess of the relative transformation. Measurement points of scan **A** are denoted by “.”; measurement points of scan **B** are denoted by “×”.

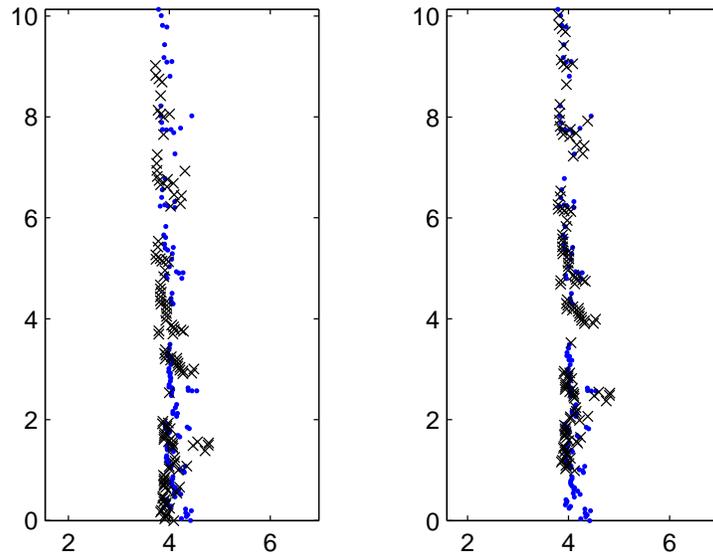


Figure 3.11. Results of segment 1 registration. Left: registration using only segment 1 of scan **A** and segment 1 of scan **B**. Right: registration using the whole scans of **A** and **B**.

### Correspondence Finding Ambiguity

Because of *sparse* and *featureless* data issues, precisely estimating the relative transformation and its corresponding distribution is difficult and the ambiguity is hard to avoid in practice. However, as long as the ambiguity is modelled correctly, this ambiguity can be reduced properly when more information or constraints are available. If the distribution does not describe the situation properly, data fusion can not be done correctly even if the incoming measurements contain rich information or constraints to disambiguate the

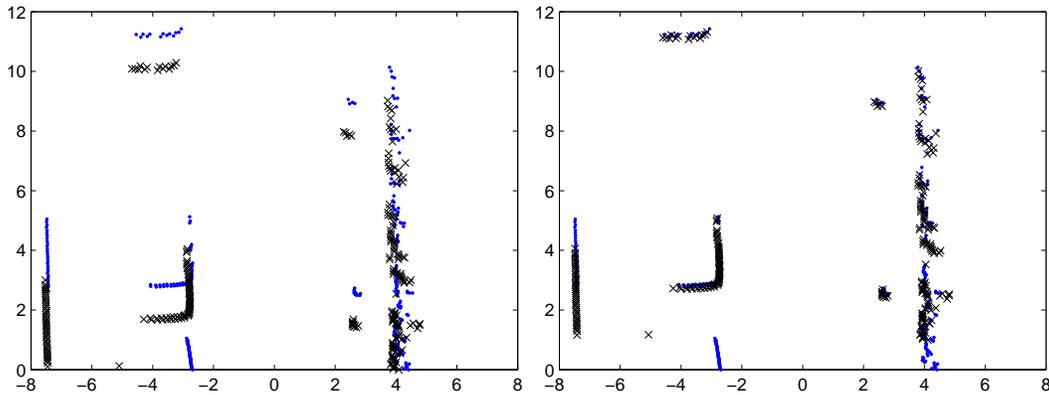


Figure 3.12. Registration results of Figure 3.11 are shown with the whole scans. Left: registration using segment 1 of scan **A** and segment 1 of scan **B**. Right: registration using the whole scans of **A** and **B**.

estimates. Therefore, although more computational power is needed, a sampling-based approach is applied to deal with the issues of correspondence finding ambiguity, or data association in the small.

Instead of using only one initial relative transformation guess, the registration process is run  $N$  times with randomly generated initial relative transformations. Figure 3.13 shows the sampling-based registration of scan segment 1 in the previous example. 100 randomly generated initial relative transformation samples are shown in the left figure and the corresponding registration results are shown in the right figure. Figure 3.13 shows that one axis of translation is more uncertain than the other translation axis and the rotation axis. Figure 3.14 shows the corresponding sample means and covariances using different numbers of samples. The covariance estimates from the sampling-based approach describe the distribution correctly.

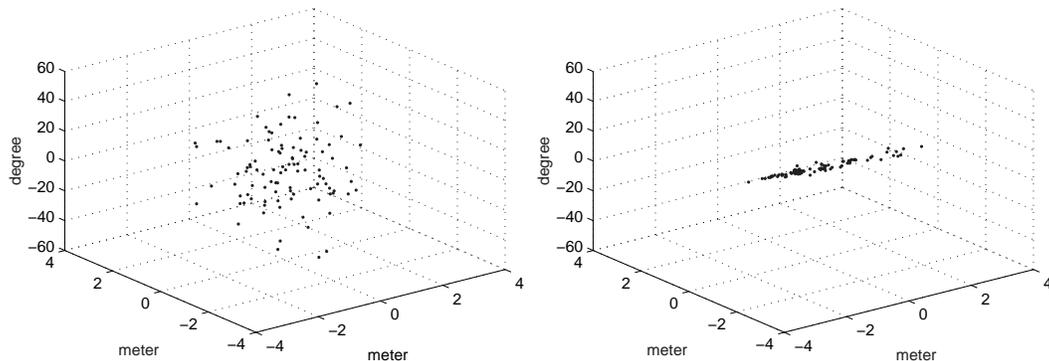


Figure 3.13. Sampling-based uncertainty estimation. Left: the randomly generated initial transformation samples. Right: the transformation estimates after applying the registration algorithm.

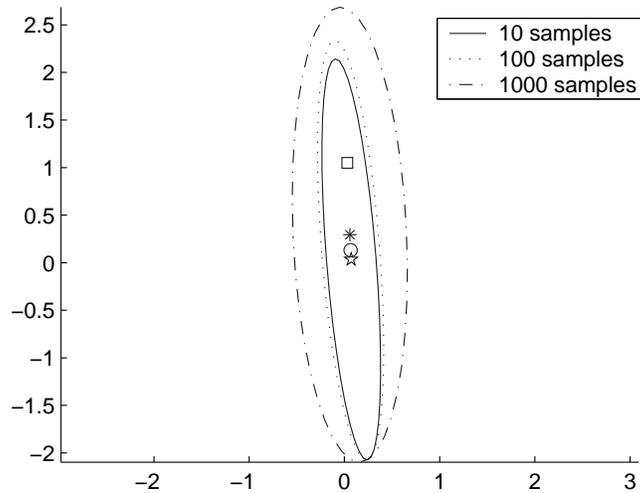


Figure 3.14. The corresponding sample means and covariances using different numbers of samples. Covariances are shown by  $2\sigma$  ellipses (95% confidence).  $\square$  is the pose estimate using the whole scans, which can be treated as the ground truth. The means estimates from 10, 100 and 1000 samples are labelled as a pentagram, a circle and a star respectively.

### Measurement Noises

Because the sampling-based approach does not handle the measurement noise issues, the grid-based method (Elfes, 1988, 1990) and the correlation-based method (Konolige and Chou, 1999) are applied and integrated for taking measurement noise into account.

First, measurement points and their corresponding distributions are transformed into occupancy grids using the perception model described in Section 3.2. Let  $g_a$  be an object-grid built using the measurement  $\mathbf{A}$  and  $g_a^{xy}$  be the occupancy of a grid cell at  $\langle x, y \rangle$ . The grid-based approach decomposes the problem of estimating the posterior probability  $p(g | \mathbf{A})$  into a collection of one-dimensional estimation problems,  $p(g^{xy} | \mathbf{A})$ . A common approach is to represent the posterior probability using log-odds ratios:

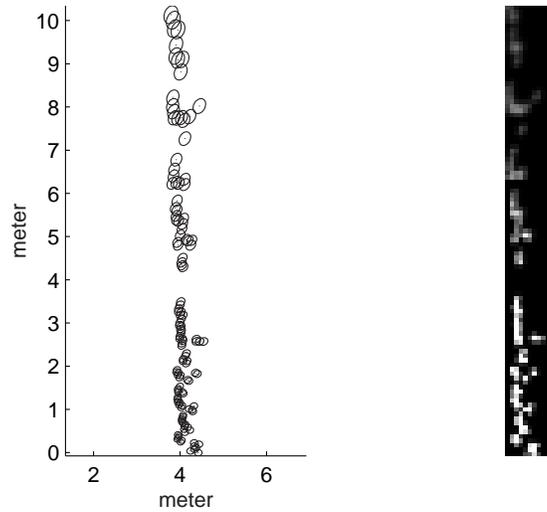
$$l_a^{xy} = \log \frac{p(g_a^{xy} | \mathbf{A})}{1 - p(g_a^{xy} | \mathbf{A})} \quad (3.8)$$

Figure 3.15 and Figure 3.16 show the corresponding occupancy grids of the segment 1 of scan  $\mathbf{A}$  and scan  $\mathbf{B}$ .

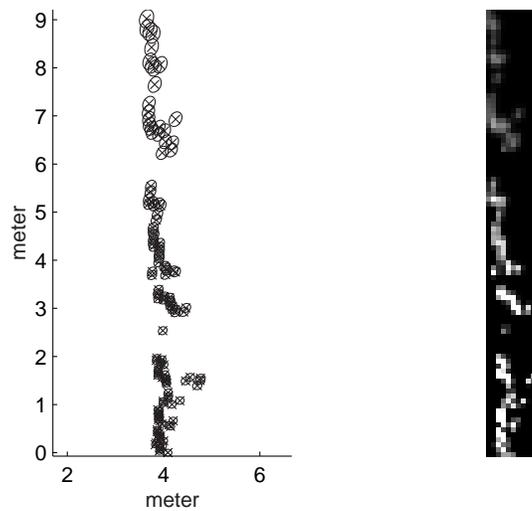
After the grid maps  $l_a$  and  $l_b$  are built, correlation of  $l_a$  and  $l_b$  is used to evaluate how strong the grid-maps are related. The correlation is computed as:

$$\sum_{xy} p(\mathbf{A}^{xy})p(\mathbf{B}^{xy}) \quad (3.9)$$

Because the posterior probability is represented using log-odds ratios, multiplication of probabilities can be done using additions.



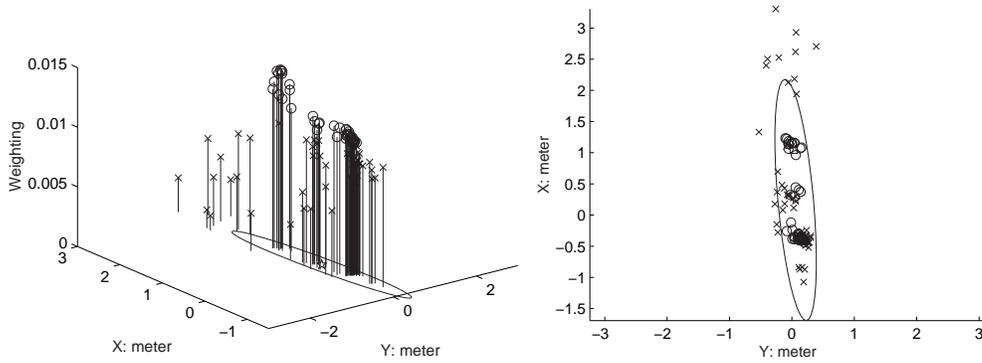
**Figure 3.15.** Occupancy grids. Left: the measurement points and their corresponding distributions of Segment 1 of **A**. Right: the corresponding occupancy grids of Segment 1 of **A**. The whiter the grid map the more certain the occupancy probability.



**Figure 3.16.** Occupancy grids. Left: the measurement points and their corresponding distributions of Segment 1 of **B**. Right: the corresponding occupancy grids of Segment 1 of **B**.

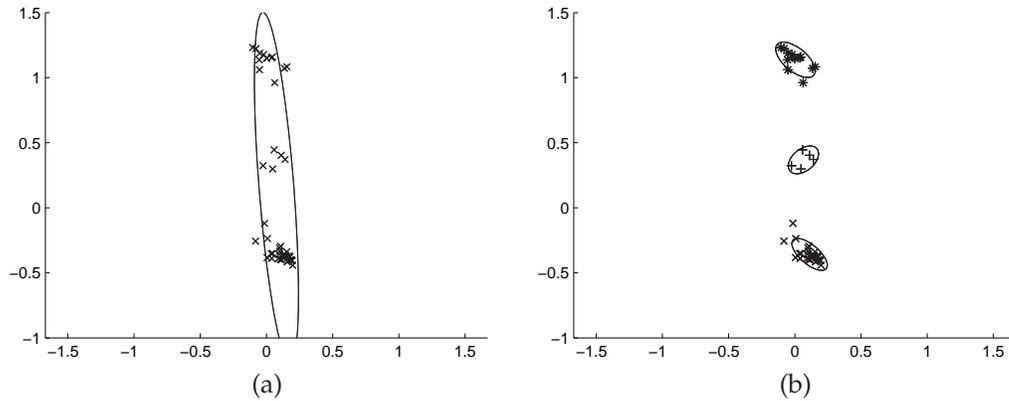
In the previous section, the sampling-based approach treated the samples equally. Now the samples are weighted with their normalized correlation responses. Figure 3.17 shows the normalized correlation responses.

The samples with low correlation responses can be filtered out for getting more accurate sample mean and covariance. Further, by properly clustering the samples, the distribution can be more precisely described by several Gaussian distributions instead of one



**Figure 3.17.** The normalized correlations of the samples. Left: 3D view. Right 2D view. The  $2\sigma$  ellipse denotes the *unweighted* sample covariance. The samples, which have correlation higher than the correlation median, are labelled by  $\circ$ . The other samples are labelled by  $\times$ .

Gaussian distribution. Figure 3.18 shows that the samples with high correlation values are clustered into three clusters and the distribution of the pose estimate now can be represented by three Gaussian distributions.



**Figure 3.18.** Mean and covariance estimation by clustering. Left: Mean and covariance estimates using the samples with high correlation values. Right: The mean and the distribution are described by three Gaussian distributions.

Based on this observation, instead of using the Particle Filter with hundreds or thousands of particles for dealing with the non-Gaussian distribution issues, we can use the proper number of samples (particles) to correctly describe non-Gaussian distributions without losing accuracy with this data-driven approach. This will be left for future work.

### Object Saliency Score

(Stoddart et al., 1996) defined a parameter called registration index, which provides a simple means of quantifying the registration error when aligning a particular shape. Similarly, we define a parameter called *object saliency score* using the trace of the autocovariance

estimate from the sampling and correlation based approach. The autocovariance is computed as:

$$p(\mathbf{A} \mid \mathcal{T}'_{[i]}, \mathbf{A}) \quad (3.10)$$

where  $\mathcal{T}'_{[i]}$  is the  $i$ th randomly generated perturbation.

The object saliency score is defined and computed as:

$$\mathbb{S} = \frac{1}{\text{trace}(\Sigma_{\mathbf{A}})} \quad (3.11)$$

where  $\mathbb{S}$  is the object saliency score, and  $\Sigma_{\mathbf{A}}$  is the autocovariance matrix of the object  $\mathbf{A}$  from the sampling and correlation based approach. The larger the object saliency score the more certain the pose estimate from registration process. Table 3.2 shows the object saliency scores of the different objects shown in Figure 3.9.

Table 3.2. Object saliency scores of the different objects shown in Figure 3.9.

Object	Object Saliency Score
Segment 1 (Bush Area)	2.1640
Segment 16 (Parked Car)	15.6495
Segment 17 (Building Wall)	9.0009
Whole scan	15.8228

According to the object saliency scores, the pose estimates of the bush area and the wall object are more uncertain than the parked car and the whole sensed area. Regardless of the initial relative transformation guess, this is intuitively correct because the whole scan and the parked car contain salient features but the bush area and the wall do not. Assuming that the environment is static, it is suggested that the whole scan should be used in registration process because the whole scan is more likely to contain salient features than individual segments and more certain pose estimates can be obtained from the registration process.

### 3.4. Hierarchical Object-based Representation for Tracking

There is a wide variety of moving objects in urban and suburban environments such as pedestrians, animals, bicycles, motorcycles, cars, trucks, buses and trailers. The critical requirement for safe driving is that all such moving objects be detected and tracked correctly. Figure 3.19 shows an example of different kinds of moving objects in an urban area. The hierarchical object representation is suitable and applicable for our applications because *free-form* objects are used without predefining features or appearances.

Because the number of measurement points belonging to small moving objects such as pedestrians is often less than four, the centroid of the measurement points is used as

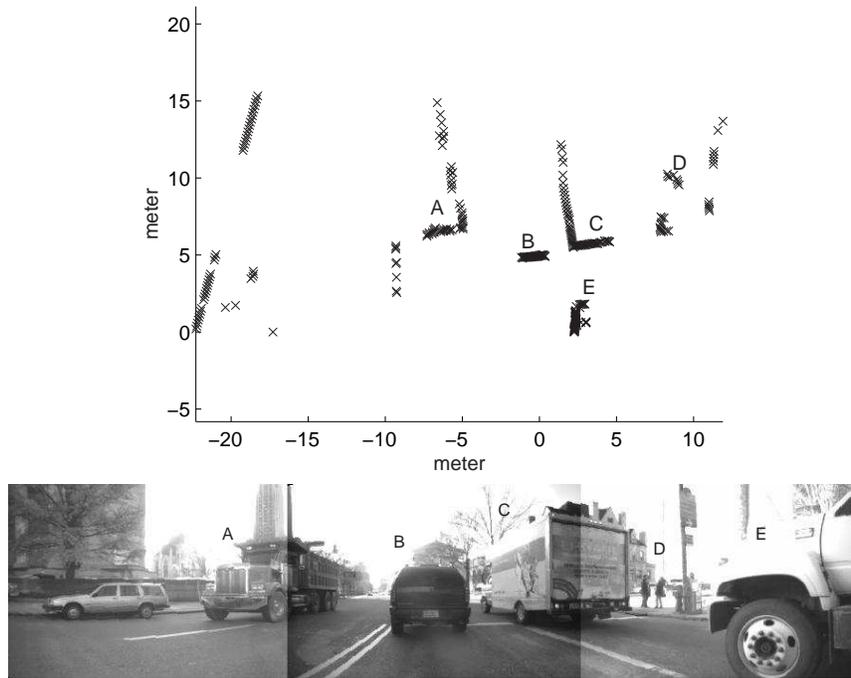


Figure 3.19. A wide variety of moving objects in urban areas. A: a dump truck, B: a car, C: a truck, D: two pedestrians, E: a truck.

the state vector of the moving object. The state vector, or *object-feature* of a small moving object contains only location without orientation because the geometrical information is insufficient to correctly determine orientation.

However, when tracking large moving objects, using the centroid of the measurements is imprecise. Different portions of moving objects are observed over different time frames because of motion and occlusion. This means that the centroids of the measurements over different time frames do not present the same physical point. Figure 3.20 shows the different portions of a moving car observed over different time frames.

Therefore, the sampling and correlation based range matching algorithm is used to estimate the relative transformation between the new measurement and the *object-grids* and its corresponding distribution. Because the online learned motion models of moving objects may not be reliable at the early stage of tracking, the predicted location of the moving object may not good enough to avoid the local minima problem of the ICP algorithm. Applying the sampling- and correlation-based range matching algorithm to correctly describe the uncertainty of the pose estimate is especially important.

Since the big object orientation can be determined reliably, the state vector, or object-feature, can consist of both location and orientation. In addition, the geometrical information is accumulated and integrated into the object-grids. As a result, not only are motions

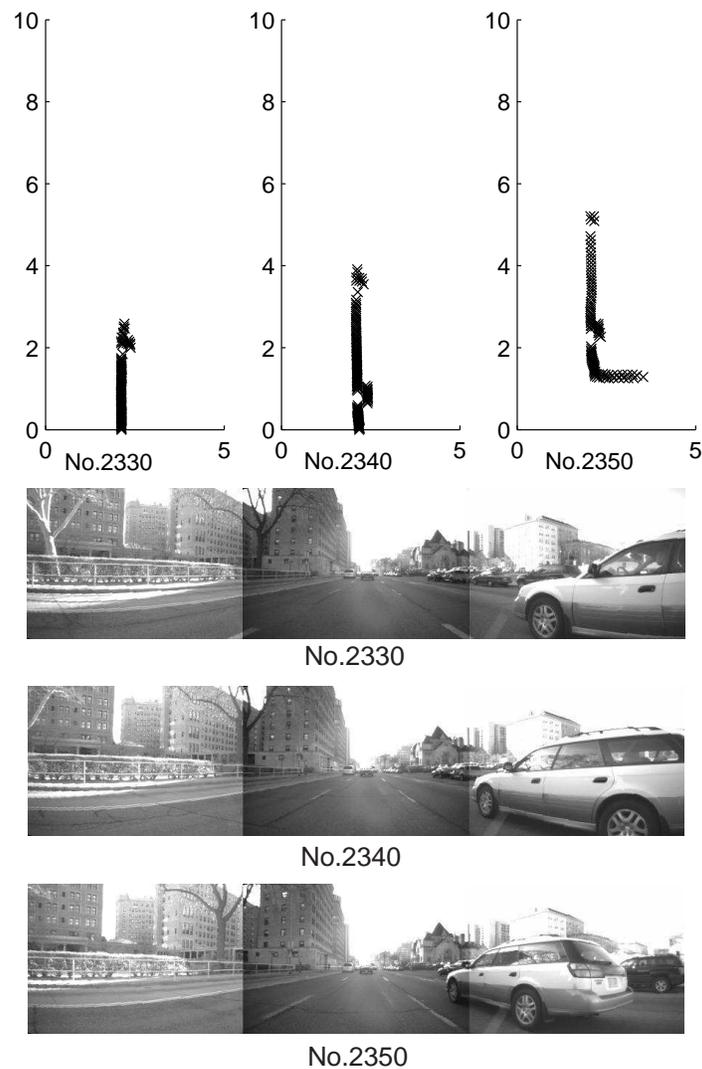
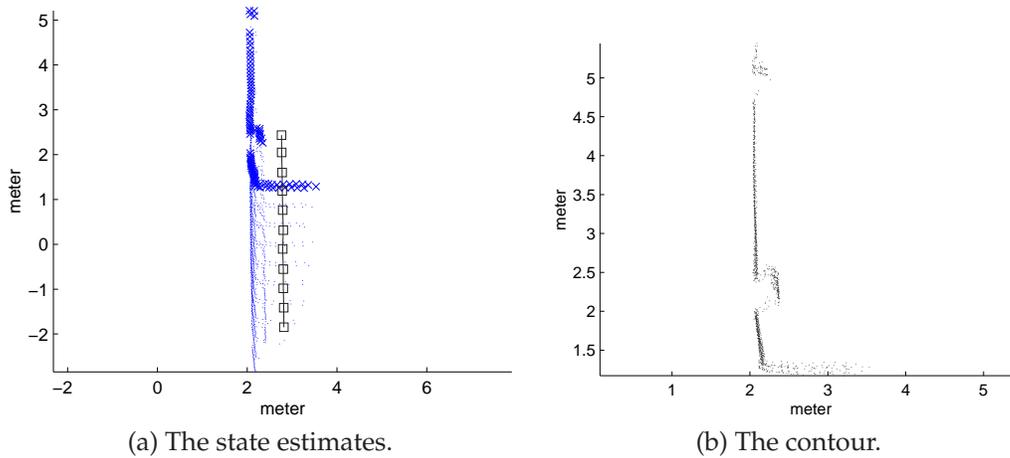


Figure 3.20. Different portions of a moving car.

of moving objects learned and tracked, but their contours are also built. Figure 3.21 shows the registration results using the SCRIM algorithm. In Chapter 5, using geometrical information to aid data association will be addressed.

### 3.5. Hierarchical Object-based SLAM

Recall that the key issues for successfully implementing SLAM in very large environments are computational complexity, representation and data association. This work does not focus on the computational complexity issue since the recent work about this issue provides several promising algorithms. Instead, we intended to develop practical and reliable algorithms for solving the representation and data association issues. In this section,



**Figure 3.21.** Registration results of the example in Figure 3.20 using the SCRIM algorithm. On the right: the states are indicated by  $\square$ , and the final scan points are indicated by  $\times$ . On the left: the scan measurements over several frames are transformed to the same coordinate system using the registration results.

we will demonstrate that *city-sized* SLAM is feasible by using the hierarchical object based approach where SLAM is accomplished *locally* using grid-based approaches and *globally* using feature-based approaches. The data association issues will be addressed in Chapter 5.

Figure 3.22 shows an aerial photo of the CMU neighborhood where the data was collected. The line indicates the Navlab11 trajectory. Figure 3.23 shows the pose estimates from the onboard inertial measurement system. It illustrates that, even using high-end inertial measurement systems, the error of the robot pose estimate accumulates and mapping using these estimates will diverge eventually. This data set contains about 36,500 scans. The visual images collected from the onboard three-camera system are only for visualization.

### Local Mapping using Grid-based approaches

Since feature extraction is difficult and problematic in outdoor environments, we apply grid-based approaches for building the map. However, as addressed in the beginning of this chapter, the grid-based approaches need extra computation for loop-closing and all raw scans have to be used to generate a new global consistent map, which is not practical for online city-sized mapping. Therefore, the grid-map is only built locally.

After localizing the robot using the sampling and correlation based range image matching algorithm, the new measurement is integrated into the grid map. The Bayesian recursive formula for updating the grid map is computed by: (See (Elfes, 1988, 1990) for a



Figure 3.22. Aerial photo of the CMU neighborhood. The line indicates the trajectory of Navlab 11.

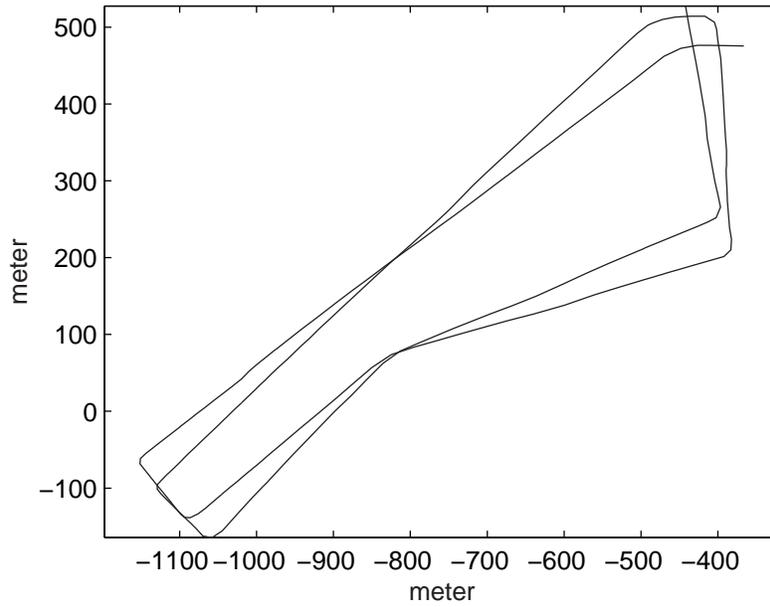


Figure 3.23. Pose estimates from the inertial measurement system.

derivation.)

$$\begin{aligned}
 l_k^{xy} &= \log \frac{p(g^{xy} | Z_{k-1}, z_k)}{1 - p(g^{xy} | Z_{k-1}, z_k)} \\
 &= \log \frac{p(g^{xy} | z_k)}{1 - p(g^{xy} | z_k)} + l_{k-1}^{xy} + l_0^{xy}
 \end{aligned} \tag{3.12}$$

where  $g$  is the grid map,  $g^{xy}$  be the occupancy value of a grid cell at  $\langle x, y \rangle$ ,  $l$  is the log-odds ratio, and

$$l_0^{xy} = \log \frac{p(g^{xy})}{1 - p(g^{xy})} \quad (3.13)$$

Practically, there are two requirements for selecting the size and resolution of grid maps: one is that a grid map should not contain loops, and the other is that the quality of the grid map should be maintained at a reasonable level. For solving the above example, the width and length of the grid map are set as 160 meters and 200 meters respectively, and the resolution of the grid map is set at 0.2 meter. When the robot arrives at the 40 meter boundary of the grid map, a new grid map is initialized. The global pose of the map and the corresponding distribution is computed according to the robot's global pose and the distribution. Figure 3.24 shows the boundaries of the grid maps generated along the trajectory using the described parameters. Figure 3.25 shows the details of the grid maps, which contain information from both stationary objects and moving objects. The details of dealing with moving objects will be addressed in the following chapters.

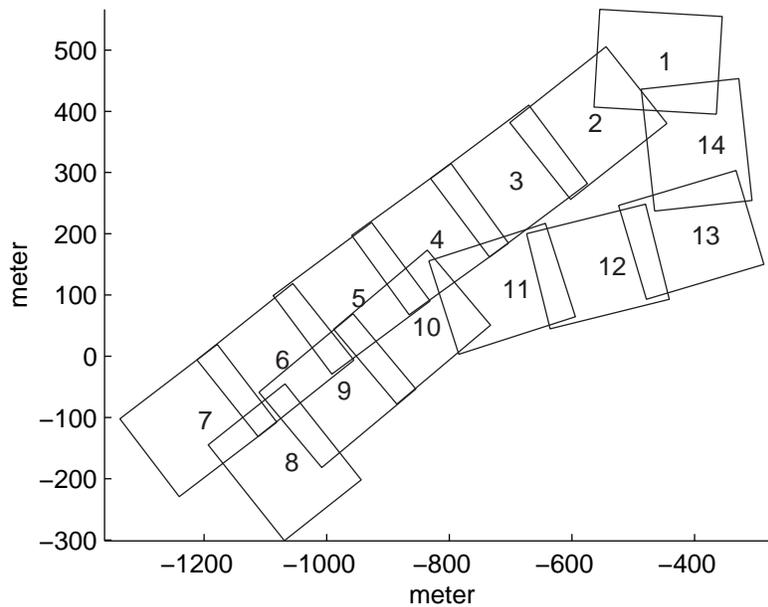
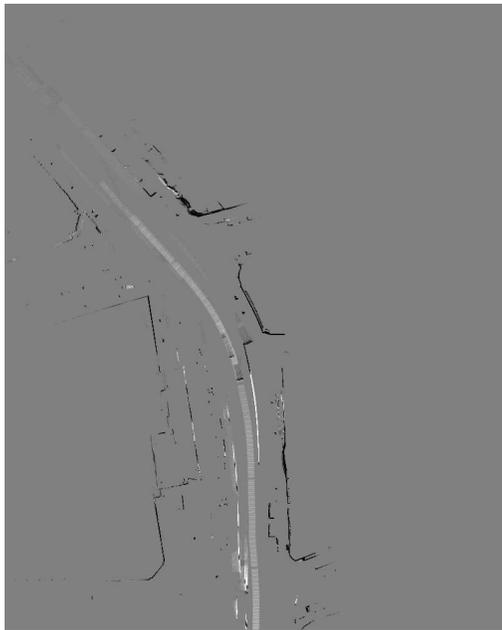


Figure 3.24. Generated grid maps along the trajectory. The boxes indicate the boundaries of the grid maps.

### Global Mapping using Feature-based Approaches

The first step to solve the loop-closing problem is to robustly *detect* loops or *recognize* the pre-visited areas. It is called the *data association in the large* problem in this dissertation or the *revisiting* problem (Stewart et al., 2003). Figure 3.26 shows that the robot entered the



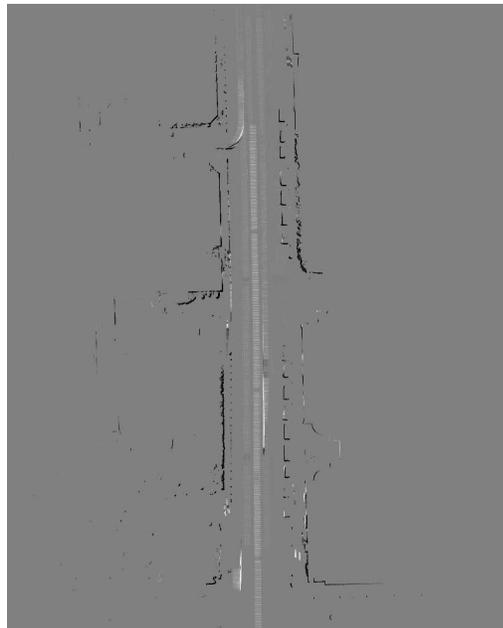
(a) Grid-map 1



(b) Grid-map 2



(c) Grid-map 3



(d) Grid-map 4

**Figure 3.25.** Details of the grid maps. *Gray* denotes areas which are not occupied by both moving objects and stationary objects, *whiter than gray* denotes the areas which are likely to be occupied by moving objects, and *darker than gray* denotes the areas which are likely to be occupied by stationary objects.

explored area. Because of the accumulated pose error, the current grid map is not consistent with the pre-built map. In this section we assume that loops are correctly detected.

The issues and solutions of the data association in the large problem will be addressed in Chapter 5.3.

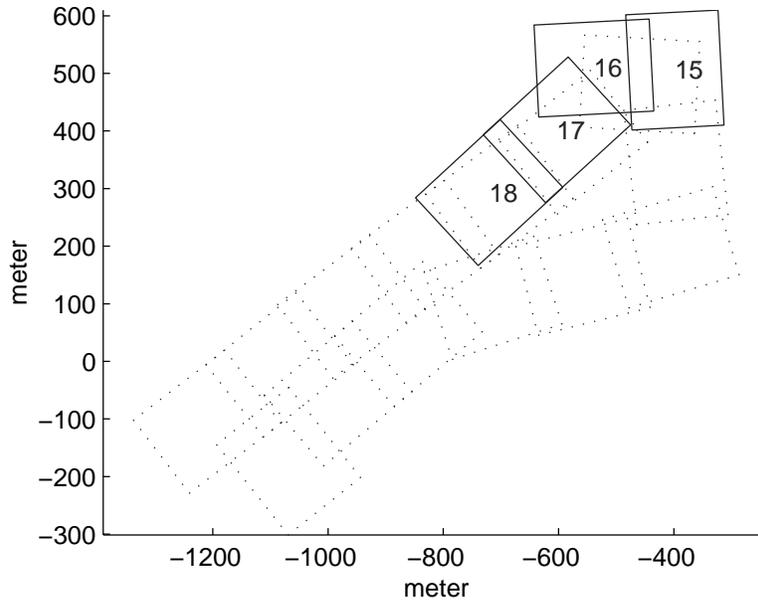


Figure 3.26. The data association in the large problem.

For closing loops in real time, feature-based approaches are applied. Because the occupancy grid approach is used for local mapping, we have to develop a method to transform or decompose the occupancy grid map into stable regions (features) and a covariance matrix containing the correlation among the robot and the regions. Unfortunately, this is still an open question. Therefore, instead of decomposing the grid maps, we treat each grid map as a 3 degree-of-freedom feature directly. Figure 3.27 shows the result without loop-closing and Figure 3.28 shows the result using the feature based EKF algorithm for loop-closing. Information from moving objects is filtered out in both figures. The covariance matrix for closing this loop only contains 14 three degree-of-freedom features.

Since we set the whole grid maps as features in the feature-based approaches for loop-closing, the uncertainty *inside* the grid maps is not updated with the constraints from detected loops. Although Figure 3.28 shows a satisfying result, the coherence of the overlay between grid maps is not guaranteed. Practically, the inconsistency between the grid-maps will not effect the robot's ability to perform tasks. Local navigation can use the current built grid map which contains the most recent information about the surrounding environment. Global path planning can be done with the global consistent map from feature-based approaches in a topological sense. In addition, the quality of the global map can be improved

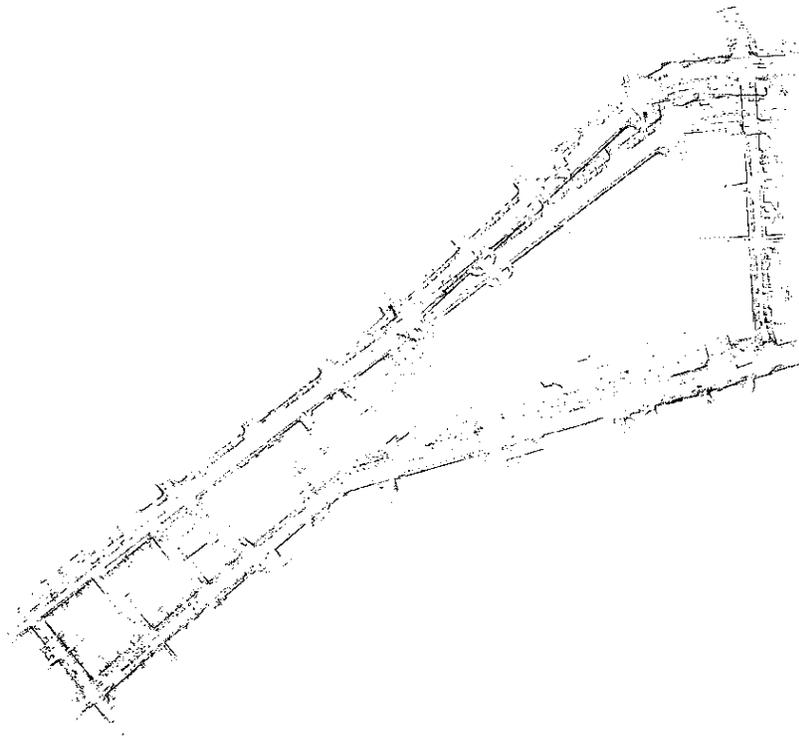


Figure 3.27. The result without loop-closing. Information from moving object is filtered out.

by using smaller grid maps to smooth out the inconsistency between grid maps. At the same time, the grid-maps should be big enough to have high object saliency scores in order to reliably solve the data association problem in the large.

### 3.6. Summary

In this chapter we have discussed the problems of perception modelling in both SLAM and moving object tracking. Three main paradigms of representation: direct methods, grid-based approaches and feature-based approaches are integrated into the framework of this hierarchical object based approach. We have presented a sampling- and correlation-based range image matching algorithm for estimating the uncertainty of the pose estimate precisely by taking the correspondence errors and measurement noise into account. The object saliency score is defined and we will describe how to use this score to aid data association in Chapter 5. We have demonstrated that the hierarchical object based representation satisfies the requirements of *uncertainty management*, *sensor characteristics*, *environment representability*, *data compression* and *loop closing mechanism*. The use of this algorithm has been verified by the experimental results using data collected from Navlab11.

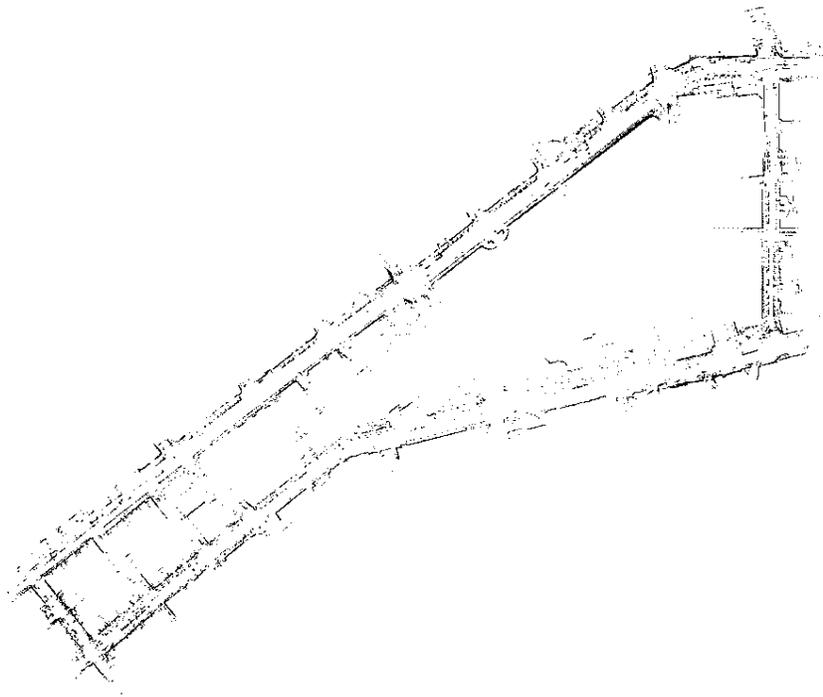


Figure 3.28. The result with loop-closing. Information from moving object is filtered out.

In the next chapter, we will describe one of the most important topics in Bayesian based SLAM and moving object tracking, *motion modelling*.



## CHAPTER 4

---

### Motion Modelling

*Make things as simple as possible but not simpler.*

– A. Einstein

**M**OTION MODELLING, or estimation of structural parameters of a system, is called *system identification* in the control literature and *learning* in the artificial intelligence literature. From a theoretical point of view, motion modelling is as important as perception modelling in Bayesian approaches. From a practical point of view, without reasonably good motion models, the predictions may be unreasonable and cause serious problems in data association and inference. Reliable prediction of both robot and moving objects is a key for collision warning, static and dynamic obstacle avoidance, and planning.

Because the robot motion model can be learned or identified in advance, it is reasonable to assume that the robot motion model is known and the only uncertainties consist of additive noises with known statistical properties. In the SLAM problem, motion modelling is easier than perception modelling. In the moving object tracking problem, motion modes of a moving object are partially unknown and possibly time varying in practice, making motion modelling difficult.

We begin this chapter with a short introduction of model selection. We will introduce the robot motion modelling problem briefly, and the rest of this chapter will focus on the issues of moving object motion modelling.

#### 4.1. Model Selection

The performance of offline learning or online adapting algorithms strongly relates to their prediction capability, which highly depends on the selected models (Hastie et al., 2001). Because model selection is very important for both robot and moving object motion modelling, in this section we introduce the fundamentals briefly.

### Off-line Learning

Intuitively, complex models are better than simple models because complicated ones can describe complicated systems more precisely than simple ones. It is true in the training stage of learning. In the training stage, training error consistently decreases with model complexity. However, training error is not a good index for testing error. For offline learning, a model with zero *training* error is often overfitted and does not work well in the general case. Figure 4.1 shows that in the *testing* stage more complicated models are able to adapt more complicated underlying structures, but the estimation error increases. Estimation error also increases when the model is simpler. There is an *optimal model complexity* between simple and complicated models. See Chapter 8 of (Hastie et al., 2001) for more details about the model selection and model assessment related issues.

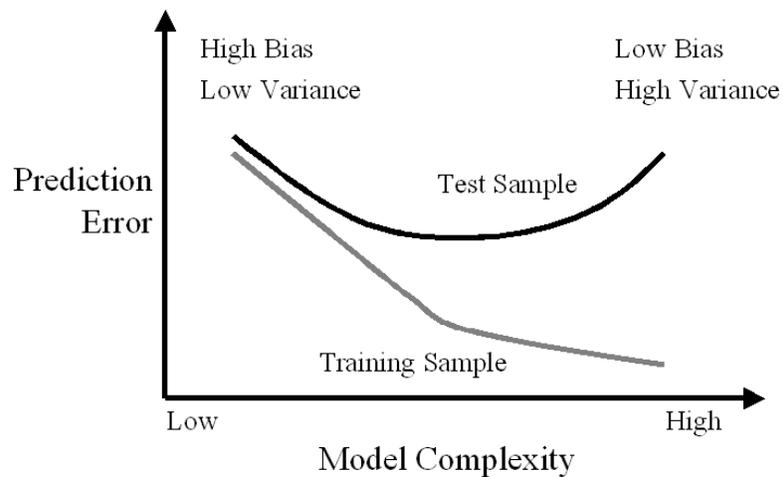


Figure 4.1. Model complexity. Behavior of test sample and training sample errors as the model complexity varies, adapted from (Hastie et al., 2001)

### Online Adapting

For online adapting, using more models is not necessarily the optimal solution. Additionally, it increases computational complexity considerably. (Li and Bar-Shalom, 1996) provided a theoretical proof that even the optimal use of motion models does not guarantee better tracking performance.

Use of a fixed set of models is not the only option for multiple model based tracking approaches. A variable structure (VS) can be used in multiple model approaches (Li and Bar-Shalom, 1992; Li, 2000). By selecting the most probable model *subset*, estimation performance can be improved. However, this requires more complicated computation procedures.

For tracking, not only motion but also other types of information or constraints can be selected and added to the model set. In (Kirubarajan et al., 1998), terrain conditions are used as constraint models and are added to the model set to improve performance of ground target tracking via a VS-IMM algorithm.

## 4.2. Robot Motion Modelling

Robot motion modelling involves kinematics and dynamics, which are the core topics in the control literature. This topic has received a great deal of attention over several decades in the car, air, and marine industries. Kinematic equations describe the spatial relationships between the vehicle (robot) frame and the world frame. Dynamic equations are derived from Newtonian or Lagrangian mechanics, which explain the force interaction between the robot and its surrounding environment. Research on manned vehicles has been used as the guideline for robot design and modelling. For robots operated on the ground, (Wong, 2001) provides a comprehensive treatment of ground vehicle modelling on both road and off-road terrains. For marine robots, (Fossen, 2002) is the definitive textbook and (Wang, 1996) addresses the modelling and control issues of underwater vehicles. For aerial vehicles, (McLean, 1990) is a good reference.

One of the keys for online robot motion modelling is modelling of motion sensors. With advanced sensor modelling, accurate relative localization using inexpensive odometry has been demonstrated to be feasible for mobile robots in (Kelly, 2000) and (Doh et al., 2003).

While perception modelling has been the subject of extensive study, only few results such as (Julier and Durrant-Whyte, 2003) address the role played by the robot motion model in the navigation system. Although robot motion modelling is beyond the scope intended by this dissertation and will not be discussed further, a good treatment of robot motion modelling can be a crucial factor for robots operating at high speeds, on three-dimensional rough terrain, in the sky, or underwater.

## 4.3. Moving Object Motion Modelling

Because of their low computational cost and satisfactory performance, the IMM algorithm and its variants (Mazor et al., 1998) have been successfully implemented in many tracking applications for dealing with the moving object motion modelling problem. Since the formulation of the IMM algorithm has been addressed in Section 2.3, we will only describe the motion models used in the IMM algorithm. Because it is very time-consuming and difficult to obtain ground-truth for moving object tracking in our applications, we use synthetic data to examine the algorithms under varying conditions.

### Discrete Time State Space Model

Just as with the robot motion modelling problem, moving objects can be described by kinematic and dynamic equations. Motion models used for moving object tracking often have a simpler form because of *limited data* associated with moving objects and *limited time* for motion model learning or adapting in practice. In the tracking literature, moving objects can be characterized by a simplified discrete time state space model, namely:

$$\mathbf{o}_{k+1} = F_k \mathbf{o}_k + G_k \mathbf{u}_k + \mathbf{v}_k \quad (4.1)$$

where  $\mathbf{o}_k$  is the state vector at time  $k$ ,  $F_k$  is the transition matrix of the system at time  $k$ ,  $G_k$  is the discrete time gain at time  $k$ ,  $\mathbf{u}_k$  is the input assumed to be constant over a sampling period, and  $\mathbf{v}_k$  is the discrete time process noise. Note that Equation 4.1 describes a time-varying discrete time system.

In most moving object tracking applications, the control input  $u_k$  is *unknown* and there is no sensor measuring the control input directly. One approach for dealing with this problem is to directly estimate this unknown input using the measurements associated with moving objects. However, (Fitzgerald, 1980) has showed that it is often not feasible to obtain an accurate control input estimate if only positional information is available.

Instead, by assuming that a moving object behaves according to one of a finite number of models, multiple model approaches are applied widely and successfully in many maneuvering object tracking applications. In this section, we will describe two of the basic motion models, the constant velocity model and the constant acceleration model.

#### The Constant Velocity Model

The constant velocity (CV) model is shortened from the the piecewise constant white noise constant velocity model, which assumes that the control input of the system can be described as white noise. In other words, the acceleration of the moving object is modelled as white noise.

As described in (Bar-Shalom and Li, 1995), the state equation for this model is:

$$\mathbf{o}_{k+1} = F \mathbf{o}_k + \Gamma \mathbf{v}_k \quad (4.2)$$

where the process noise  $\mathbf{v}_k$  is a zero-mean white acceleration sequence.

Let the Cartesian state vector of the system be:

$$\mathbf{o} = [x, v_x, y, v_y]^T \quad (4.3)$$

The time-invariant state transition matrix is

$$F = \begin{bmatrix} F_x & \mathbf{0} \\ \mathbf{0} & F_y \end{bmatrix}, \quad F_x = F_y = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \quad (4.4)$$

where  $t$  is the sampling period.

Assuming  $\mathbf{v}_k$  is constant during the  $k$ -th sampling period  $t$ , the increases in the velocity and the position are  $\mathbf{v}_k t$  and  $\mathbf{v}_k t^2/2$  respectively. Therefore, the noise gain is computed by:

$$\Gamma = \begin{bmatrix} \Gamma_x \\ \Gamma_y \end{bmatrix}, \quad \Gamma_x = \Gamma_y = \begin{bmatrix} \frac{1}{2}t^2 \\ t \end{bmatrix} \quad (4.5)$$

The covariance of the process noise multiplied by the gain is

$$Q = \begin{bmatrix} Q_x & \mathbf{0} \\ \mathbf{0} & Q_y \end{bmatrix} \quad (4.6)$$

where

$$\begin{aligned} Q_x = Q_y &= E[\Gamma v_k v_k \Gamma'] \\ &= \begin{bmatrix} \frac{1}{4}t^4 & \frac{1}{2}t^3 \\ \frac{1}{2}t^3 & t^2 \end{bmatrix} \sigma_v^2 \end{aligned} \quad (4.7)$$

It is suggested that  $\sigma_v$  should be of the order of the maximum acceleration magnitude  $a_M$  for the constant velocity model.

### The Constant Acceleration Model

The constant acceleration (CA) model is shortened from the piecewise constant white noise acceleration model in which the system is assumed to perform accelerations that are constant over a sampling interval. In this model, the acceleration increment is assumed to be zero-mean white sequence. The state equation of this model is the same as Equation 4.2 but the white process noise  $\mathbf{v}_k$  is the acceleration increment.

Let the Cartesian state vector of the system be:

$$\mathbf{o} = [x, v_x, a_x, y, v_y, a_y]^T \quad (4.8)$$

The transition matrix is described as:

$$F = \begin{bmatrix} F_x & \mathbf{0} \\ \mathbf{0} & F_y \end{bmatrix}, \quad F_x = F_y = \begin{bmatrix} 1 & t & \frac{1}{2}t^2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

The noise gain is

$$\Gamma = \begin{bmatrix} \Gamma_x \\ \Gamma_y \end{bmatrix}, \quad \Gamma_x = \Gamma_y = \begin{bmatrix} \frac{1}{2}t^2 \\ t \\ 1 \end{bmatrix} \quad (4.10)$$

The covariance of the process noise multiplied by the gain is

$$Q = \begin{bmatrix} Q_x & \mathbf{0} \\ \mathbf{0} & Q_y \end{bmatrix} \quad (4.11)$$

where

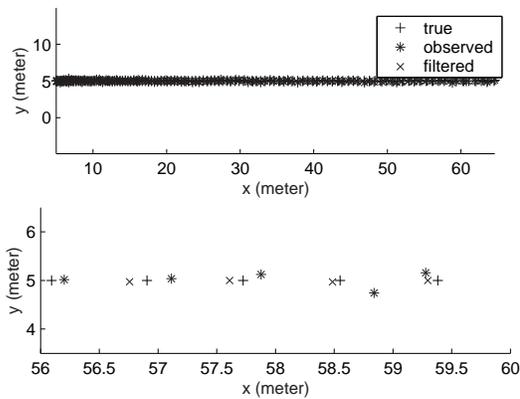
$$\begin{aligned}
 Q_x = Q_y &= \Gamma \sigma_v^2 \Gamma' \\
 &= \begin{bmatrix} \frac{1}{4}t^4 & \frac{1}{2}t^3 & \frac{1}{2}t^2 \\ \frac{1}{2}t^3 & t^2 & t \\ \frac{1}{2}t^2 & t & 1 \end{bmatrix} \sigma_v^2
 \end{aligned} \tag{4.12}$$

For the CA model, it is suggested that  $\sigma_v$  should be of the order of the magnitude of the maximum acceleration increment over a sampling period,  $\Delta a_M$ .

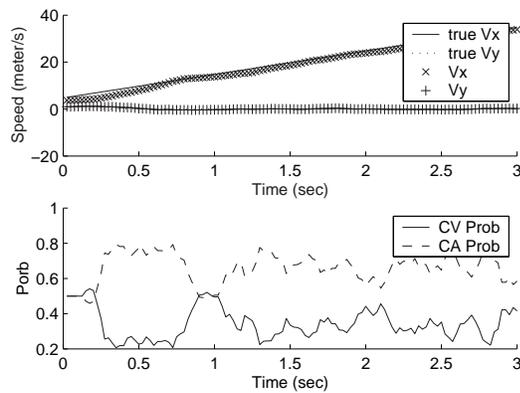
### The IMM algorithm with the CV and CA Models

In this section, we will illustrate the performance and the limitation the IMM algorithm with the CV and CA models using synthetic data. In all simulations, the sensor platform is assumed to be stationary and the moving objects are described by point-features, i.e. no orientation information. The standard deviation of the additive noise in the perception measurements is 0.1 meter.

**Case 1:** a constant acceleration maneuver. In this simulation, the initial velocity of the tracked object is 5 m/s and its acceleration is 10 m/s<sup>2</sup>. Figure 4.2 shows the tracking results, the ground truth and the measurements and the performance is satisfying. Figure 4.3 shows the velocity estimates and the probabilities of the CV model and the CA model during tracking. As expected, the CA model dominates the whole tracking process according to the probabilities.

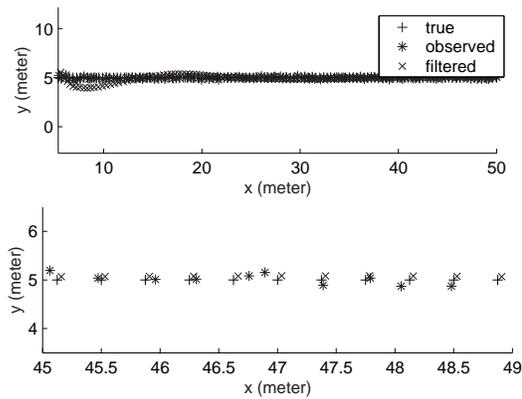


**Figure 4.2.** A simulation of the constant acceleration maneuver. Top: the whole trajectory. Bottom: a portion of the trajectory is enlarged for clarification.

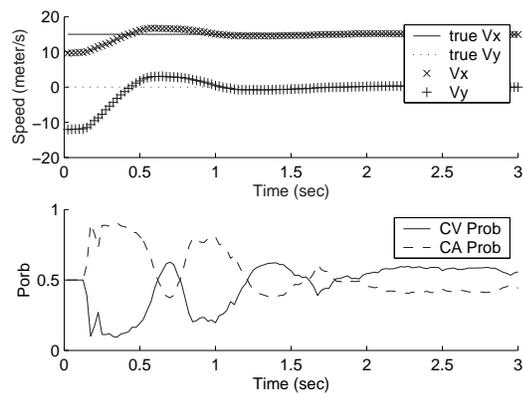


**Figure 4.3.** The velocity estimates and the probabilities of the CV model and the CA model in the constant acceleration maneuver simulation. Top: the velocity estimates. Bottom: the probabilities of the CV model and the CA model.

**Case 2: a constant velocity maneuver.** In this simulation, the velocity of this tracked object is  $15\text{ m/s}$  and its acceleration is  $0\text{ m/s}^2$ . Figure 4.4 and Figure 4.5 show the simulation results. At the beginning, the filtered state was overshooting and oscillating. After about 1 second, the system was stably tracked. According to the probabilities of the CV model and the CA model, the CV model did not strongly dominate the tracking process. The probability difference between the CV model and the CA model is not as big as the constant acceleration maneuver example. This is because of the model complexity issues, which will be discussed in Section 4.5.



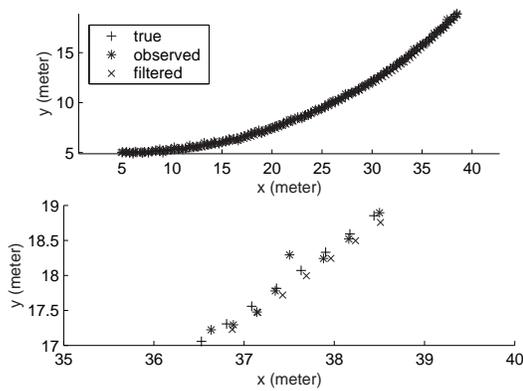
**Figure 4.4.** A simulation of the constant velocity maneuver. Top: the whole trajectory. Bottom: a portion of the trajectory is enlarged for clarification.



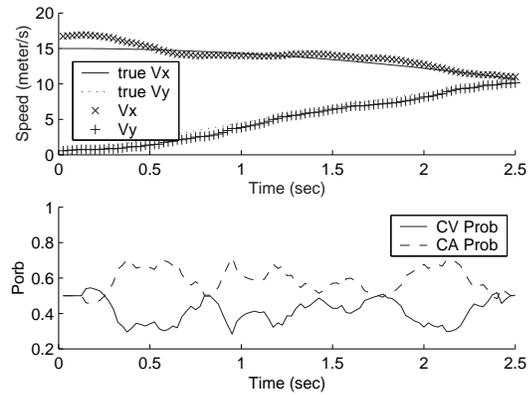
**Figure 4.5.** The velocity estimates and the probabilities of the CV model and the CA model in the constant velocity motion simulation.

**Case 3: a turning maneuver.** In this simulation, the speed of the tracked object is  $15\text{ m/s}$  and the angular velocity is  $18\text{ degree/s}$ . Figure 4.6 and Figure 4.7 show the satisfying results even the turning maneuver is neither the CV maneuver nor the CA maneuver. The CV and CA models assume that the motion of the system is independent in the  $x$  and  $y$  directions and the filter can be decoupled with respect to these axes. In the cases that a moving object performs a coordinated turn, the motion is highly correlated across these directions and the coordinated turning model can be used for getting better performance and maneuver classification.

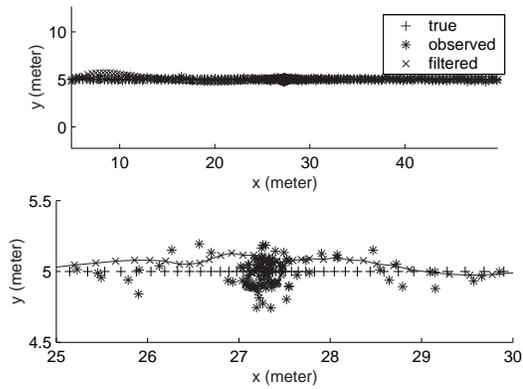
**Case 4: a move-stop-move maneuver.** This simulation demonstrates a more complicated maneuver which consists of the following motion patterns in order: constant velocity, constant deceleration, stop, constant acceleration and constant velocity. This maneuver is one of the representative maneuvers for ground moving objects in urban environments. It seems that Figure 4.8 and Figure 4.9 show the satisfying results.



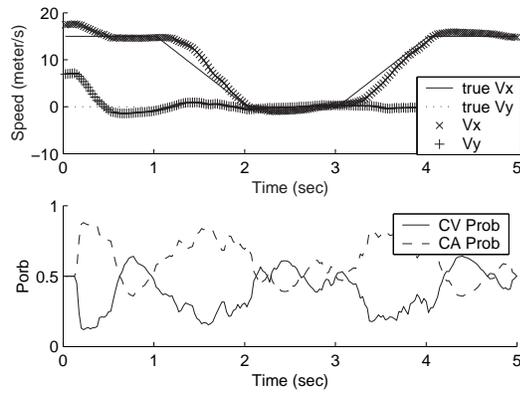
**Figure 4.6.** A simulation of the turning maneuver. Top: the whole trajectory. Bottom: a portion of the trajectory is enlarged for clarification.



**Figure 4.7.** The velocity estimates and the probabilities of the CV model and the CA model in the turning maneuver simulation.



**Figure 4.8.** A simulation of the move-stop-move maneuver. Top: the whole trajectory. Bottom: a portion of the trajectory is enlarged for clarification.



**Figure 4.9.** The velocity estimates and the probabilities of the CV model and the CA model in the move-stop-move maneuver simulation.

However, due to uncertainty in the measurements, the velocity estimate did not go to zero when the tracked object was stationary. Figure 4.8 indicates that the estimates from the IMM with CV and CA models moved around the true stationary location instead of converging to the true stationary location. Figure 4.10 shows a simulation in which a stationary object is tracked using a Kalman filter with the CV model. The filtered result overshoots and converges slowly. For SLAM with generic objects, such overshoots and slow convergence degrade inference and learning of the whole process. We will discuss this issue further in the next section.

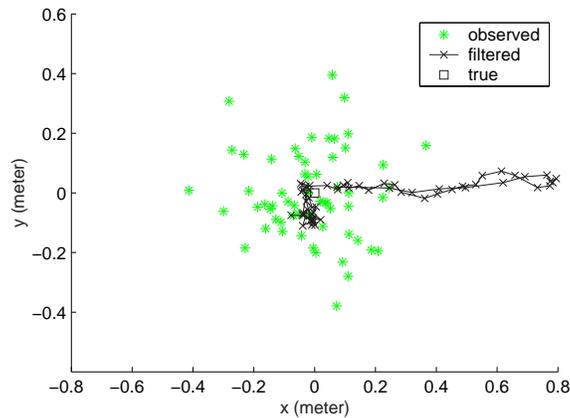


Figure 4.10. The simulation in which the stationary object is tracked using a Kalman filter with a CV model. Green \* are the measurements and red × are the filtered states.

#### 4.4. The Stationary Motion Model

For dealing with move-stop-move maneuvers, (Kirubarajan and Bar-Shalom, 2000; Coraluppi et al., 2000; Coraluppi and Carthel, 2001) have suggested adding a stationary motion model to the motion model set. In this section, we will describe two models for stationary motion modelling: one is a stop model simplified from the CV model and the other is a stationary process model. We will explain why using the Kalman filter with the stop model simplified from the CV model is not correct.

##### The Stop Model Simplified from the CV Model

Following the derivation of the CV model and the CA model, the stop model can be derived by simplifying the CV model.

Let the Cartesian state vector of the stop model be:

$$\mathbf{o} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.13)$$

The time-invariant state transition matrix of the stop model is:

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.14)$$

In the stop model,  $\mathbf{v}_k$  is zero-mean white velocity sequence. The noise gain is computed by:

$$\Gamma = \begin{bmatrix} t \\ t \end{bmatrix} \quad (4.15)$$

The covariance of the process noise multiplied by the gain is:

$$Q = \begin{bmatrix} t^2 & 0 \\ 0 & t^2 \end{bmatrix} \sigma_v^2 \quad (4.16)$$

$\sigma_v$  should be of the order of the measurement noise magnitude from both motion and perception sensors. Before describing the correctness of this stop model, we describe the second model for stationary motion modelling, the stationary process model.

### The Stationary Process Model

In the time series analysis literature, there are a number of theorems and models about stationary processes. In this section, we introduce the definitions of stationary processes briefly and describe the stationary process model used for stationary motion modelling.

It is defined that a series of observations  $z_0, z_1, z_2, \dots$  is strongly stationary or strictly stationary if

$$(z_{k_1}, \dots, z_{k_2}) = (z_{k_1+h}, \dots, z_{k_2+h}) \quad (4.17)$$

for all sets of time points  $k_1, \dots, k_2$  and integer  $h$ . A sequence is weakly stationary, or second order stationary if the expectation value of the sequence is constant and the covariance of  $z_k$  and  $z_{k+h}$ ,  $cov(z_k, z_{k+h}) = \gamma_h$ , is independent of  $k$ . A white noise sequence is a second order stationary series.

The stationary process model is assumed to be properly described by a second order stationary series. Because the motion mode of moving objects is time-varying, stationary process should be identified *locally*.

There are a number of methods to test whether a series can be considered to be white noise or whether a more complicated model is needed. Because of limited data and time in practice, the mean and the covariance of the series are used to decide if the series is a stationary process.

### Comparison

We compare the performances of these two stationary motion models. The stop model simplified from the CV model is used with Kalman filtering. In the stationary process model, the mean and covariance are computed directly and the recursive formula is derived and used.

Figure 4.11 and Figure 4.12 shows the results of stationary object tracking. Both results from the stop model and the stationary process model were not overshooting. The stationary process model converged to the true location. But the result of the stop model did not well represent the statistical properties of the stationary motion sequence.

Figure 4.13 and Figure 4.14 shows the results of constant velocity maneuver tracking using these two stationary motion models. The filtering with the stop model performed the wrong inference. Instead of taking the whole sequence into account, the Kalman filtering

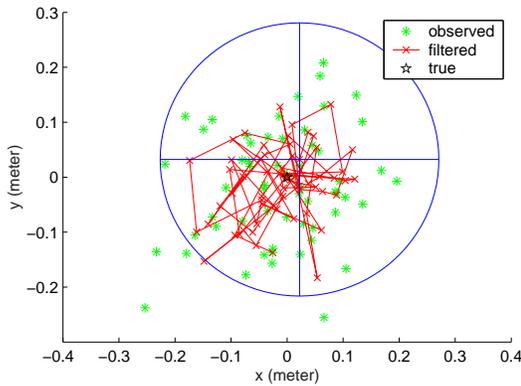


Figure 4.11. A simulation in which the stationary object is tracked using a Kalman filter with the stop model. The distribution of the last state estimate is shown by  $1\sigma$  ellipse.

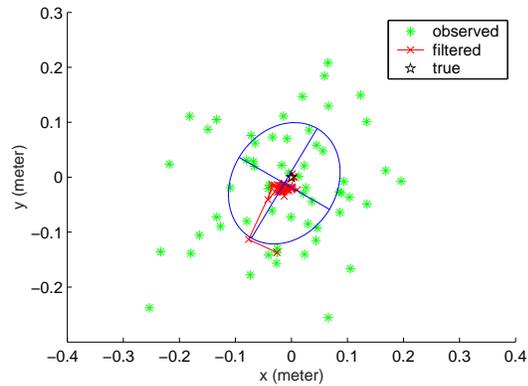


Figure 4.12. A simulation in which a stationary object is tracked using the stationary process model. The distribution of the last state estimate is shown by  $1\sigma$  ellipse.

with the stop model performs the average of the current measurement and the previous one. On the other hand, the covariance of the stationary process model represents the statistical property of the data properly, which indicates that the time series measurements can not be described by the stationary process model and this series should be a moving process.

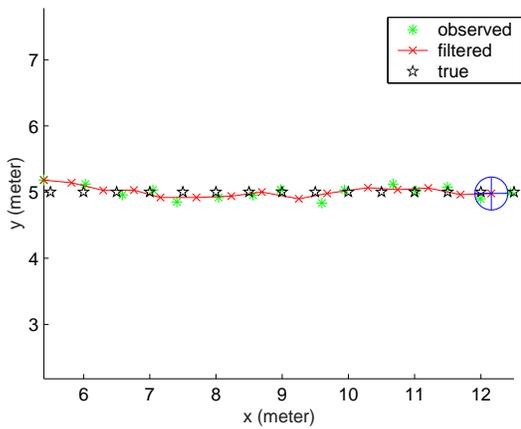


Figure 4.13. A simulation in which an object moving at a constant velocity is tracked using a Kalman filter with the stop model. The distribution of the last state estimate is shown by  $1\sigma$  ellipse.

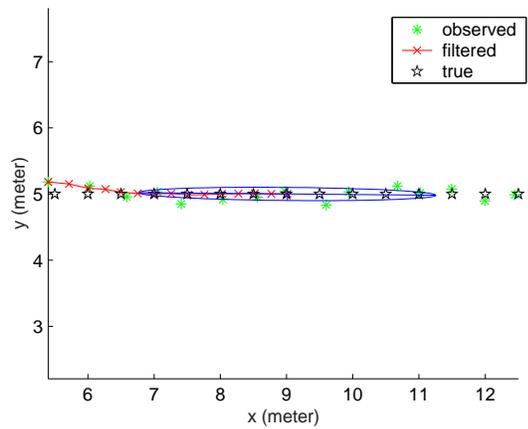


Figure 4.14. A simulation in which the constant velocity object is tracked using the stationary process model. The distribution of the last state estimate is shown by  $1\sigma$  ellipse.

As a result, the stationary process model is used for stationary motion modelling. In this section, the stop model indicates the model simplified from the constant velocity model

for distinguishing from the stationary process model. Beyond this section, the stationary process model and the stop model are exchangeable.

Compared to the CA model and the CV model, the stationary process model is an extremely simple model. In the next section, we will discuss the effects of model complexity in tracking.

#### 4.5. Model Complexity

Given a fixed set of models, the IMM algorithm performs *model averaging* but not *model selection*. It is a *decision-free* or *soft-decision* algorithm. The overall estimate is the probabilistically weighted sum of all models and the overall covariance is determined accordingly. For move-stop-move object tracking, it seems that we could just add the stationary process model to the model set and solve the problem with the same IMM algorithm. However, as observed in (Shea et al., 2000; Coraluppi and Carthel, 2001), all of the estimates tend to *degrade* when the stop model is added to the model set and mixed with other moving motion models. In this section, we attempt to provide a theoretical explanation of this phenomenon.

##### The Nested Model Set

The models used for tracking belong to a *nested* model set in which the stationary process model is a subset of the CV model and the CV model is a subset of the CA model. Figure 4.15 shows this nested model set. In the nested model set, the more complicated models can exactly describe the simpler maneuvers such as constant velocity motions and stationary motions. This explains why the CA model and the CV model have similar probability values for tracking an object moving at a constant velocity. For tracking a complicated maneuver, the CA model has much higher probability value than the CV model since the CV model can not describe this complicated maneuver well.

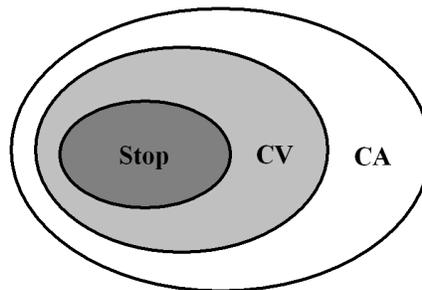


Figure 4.15. The nested model set.

Since one CA model can explain all these stop, constant velocity, and constant acceleration maneuvers, why do we need to use all these models? Assuming enough data is given and the mode of the moving object only obeys one model, more complicated models such as the CA model indeed perform better than the simpler models. Recall the model selection issues addressed in Section 4.1. In the cases with limited data and operations in real time, a complicated model is often overfit when the system is in a simple mode and more data is needed to perform inference accurately.

### Occam's Razor

Chapter 28 of (MacKay, 2003) proves that simpler models are preferred than more complicated models when models are compared using Bayes' theorem. In other words, when the models in the model set have similar probability values, Bayesian theorem prefers to choose the simpler model. It is also called *Occam's Razor*.

Figure 4.16 shows Occam's Razor embodied by Bayesian theorem. The evidence of the model  $\mathcal{M}^i$  is the probability of the data given this model,  $p(Z | \mathcal{M}^i)$ . A more powerful model such as the CA model can make larger range of predictions than simpler models such as the CV model and the stop model. However simple models provide stronger evidence than complicated models when data falls into the area that both models share. For instance, when data falls in region  $C^{CV}$ , the simpler and less powerful CV model is more probable than the CA model.

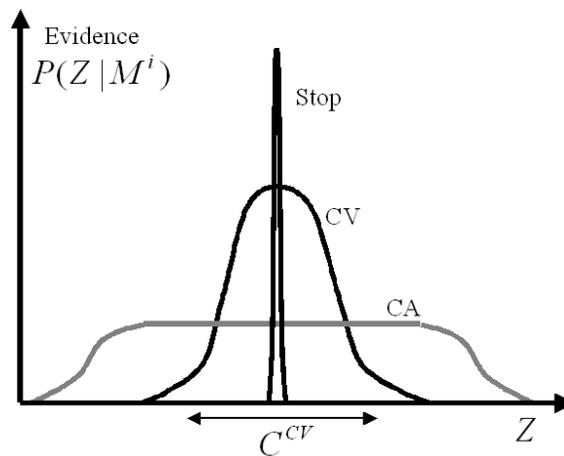


Figure 4.16. Occam's Razor: Bayesian inference automatically and quantitatively embodies Occam's Razor, modified from Chapter 28 of (MacKay, 2003)

Because the CV model and the CA model share a large portion of  $Z$  axis as shown in Figure 4.16, the degradation is not so significant when the CV model and the CA model are

mixed outside the shared region (see Table 4.1). Because the stationary process model is an *extremely* simple model which can only explain the stationary motion, a point in the axis of  $Z$ , the stop model should not be used beyond this point. When the stationary process model is added to the model set, the performance beyond this stationary point degrades as observed in (Shea et al., 2000; Coraluppi and Carthel, 2001). As a result, selecting probable models should be done with great caution, especially when dealing with very simple models such as the stationary process model. In the next section, we describe the move-stop hypothesis tracking for dealing with this issue.

**Table 4.1.** Model Complexity.  $\checkmark$  indicates that the model is perfect for the specific motion,  $\triangle$  indicates that the model is adequate for the specific motion and  $\times$  indicates that the model is wrong for the specific motion.

	The Stop model	The CV model	The CA model
Stationary Motion	$\checkmark$	$\triangle$	$\triangle$
Constant Vel. Motion	$\times$	$\checkmark$	$\triangle$
Constant Acc. Motion	$\times$	$\times$	$\checkmark$

#### 4.6. Move-Stop Hypothesis Tracking

In practice, the *minimum detection velocity* (MDV) can be obtained by taking account of the modelled uncertainty sources. For objects whose velocity estimates from the IMM algorithm with the moving models are larger than this minimum detection velocity, the objects are unlikely to be stationary and the IMM algorithm with the moving models should perform well. In the cases that the performance of the IMM algorithm with the CV model and the CA model is not satisfying, other motion models can be added to the model sets or a variable structure multiple model estimation algorithms can be applied. However they are computationally demanding.

For objects whose velocity estimates are less than this minimum detection velocity, tracking should be done with great caution. Instead of adding the stationary process model to the model set, move-stop hypothesis tracking is applied where the move hypothesis and the stop hypothesis are inferred separately.

For move hypothesis inference, tracking is done via the IMM algorithm with the CV model and the CA model. For stop hypothesis inference, the stationary process model is used to verify if the system is a stationary process at the moment with a short period of measurements. The covariances from the move hypothesis and the stop hypothesis are compared. The hypothesis with more certain estimates will take over the tracking process. Figure 4.17 and Figure 4.18 show the simulation result of move-stop hypothesis tracking. Figure 4.18 is the enlargement of Figure 4.17 around the true stationary location.

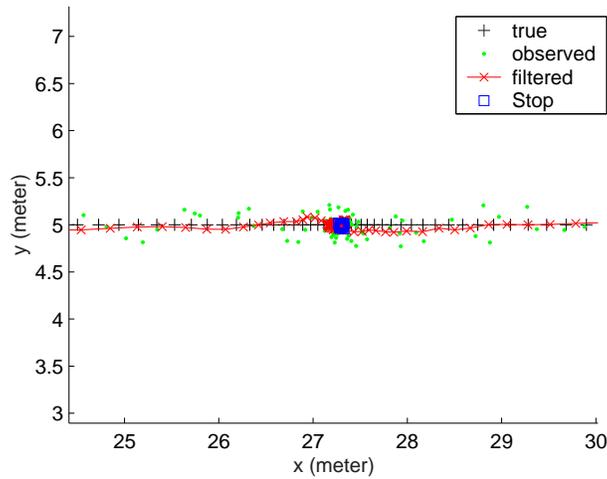


Figure 4.17. A simulation of the move-stop-move maneuver tracked by move-stop hypothesis tracking.

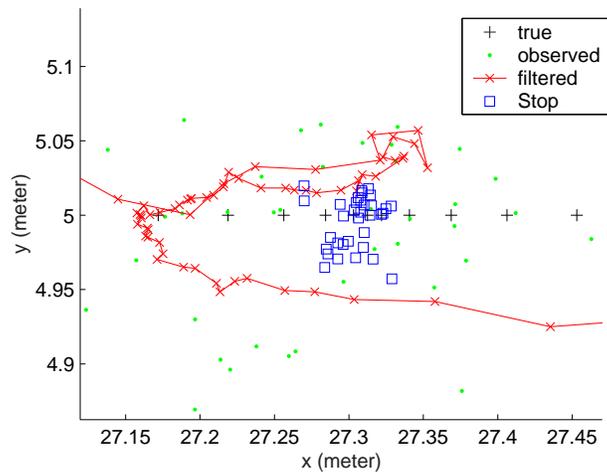


Figure 4.18. The enlargement of Figure 4.17.

Besides comparing the uncertainty estimates of the move hypothesis and the stop hypothesis, data association is a means to select the correct hypothesis. Recall that the estimates are often overshooting when the CV model or the CA model is used to track a stationary object in which correspondences between the object and new measurements usually can not be established because new measurements are outside the gates of the moving models. More details about data association will be addressed in the next chapter.

#### 4.7. Simultaneous Multiple Moving Object Tracking

Thus far, we have addressed the motion modelling issues in the single moving object tracking problem. For multiple moving object tracking, it can be decoupled and treated as

the single moving object tracking problem if the objects are moving independently. However, in many tracking applications, moving objects are moving dependently such as sea vessels or air fighters moving in formation. In the urban and suburban areas, cars or pedestrians often move in formation as well because of specific traffic conditions. Although the locations of these objects are different, velocity and acceleration may be nearly the same in which these moving objects tend to have highly correlated motions.

Similarly to the SLAM problem, the states of these moving objects can be augmented to a system state and then be tracked simultaneously. (Rogers, 1988) proposed an augmented state vector approach which is identical to the SLAM problem in the way of dealing with the correlation problem from sensor measurement errors. See Chapter 7 of (Blackman and Popoli, 1999) for details of this algorithm.

#### **4.8. Summary**

In this chapter, we discussed the importance of motion modelling in SLAM and moving object tracking from both theoretical and practical points of view. We mainly focused on the moving object motion modelling issues, in particular move-stop-move object tracking. We explained why a stop model simplified from the CV model is not correct and why a stationary process model should be used instead. Because of the model complexity effects in motion modelling, we presented the move-stop hypothesis tracking algorithm for tracking objects whose velocity estimates are smaller than the minimum detection velocity.

Thus far, the described theories and algorithms of perception modelling and motion modelling assume correct data association. Because of uncertainty, the data association problem is unavoidable in practice. In the next chapter, the data association problem will be discussed in detail.

## CHAPTER 5

---

### Data Association

*Only the paranoid survive.*

– *Andy Grove*

**D**ATA ASSOCIATION is one of the most important and challenging problems in practice. It is important because wrong data association can diverge inference and learning. It is challenging because the most probable association between measurements and objects needs to be selected from a very large set of possible matches. In this chapter, we address the data association problem in the small, in the cluttered and in the large:

- *Data association in the small, or correspondence finding*, involves determining correspondences at the measurement-to-measurement level and the segment-to-object level using two or multiple consecutive measurements. It is difficult when measurements are sparse or objects are featureless.
- *Data association in the cluttered* aims to resolve the situations in which the ambiguity can not be removed with techniques of data association in the small. This is critical for clarifying and maintaining identifications of objects in cluttered environments.
- *Data association in the large, or revisiting*, is the recognition of places where the robot is in a previously explored area. It is difficult because of accumulated pose estimate errors, unmodelled uncertainty sources, temporary stationary objects and occlusion.

With the knowledge of Chapter 3 and Chapter 4, not only *kinematic* information but also *geometric* information are used to aid data association. In many situations, measurements or environments are not informative, so that data association problem in the small can not be solved immediately even with the use of spatial, geometric and kinematic clues.

However, as long as ambiguity is modelled properly, a delayed but firm decision can be made until the later measurements are received and are enough to disambiguate the previous uncertain association. For tackling data association in the large, these techniques may be not sufficient.

In this chapter, we will address three principles to tackle these data association problems:

- Exploiting information contained in measurements,
- Modelling or describing ambiguity correctly, and
- Increasing search space to deal with unexpected situations.

We begin this chapter with the data association problem in the small.

### 5.1. Data Association in the Small

There are two levels of data association in the small: one is the measurement-to-measurement level and the other is the segment-to-object level. Since segments or objects contain either raw measurement points or a grid map, the sampling and correlation based range image registration algorithm is used for solving the measurement-to-measurement level data association.

In this section, the segment-to-object level data association is addressed. An object score function is defined and used for quantitatively measuring the confidence of objects over time. Based on Chapter 3 and Chapter 4, kinematic and geometric information is used for computing the object score function.

#### Object Score Function

Following research in the tracking literature (Sittler, 1964; Blake et al., 1999), a likelihood ratio is defined in order to evaluate segment-object hypotheses:

$$\begin{aligned} \frac{P_T}{P_F} &= \frac{p(\mathcal{H}_T | Z_k)}{p(\mathcal{H}_F | Z_k)} \\ &= \frac{p(Z_k | \mathcal{H}_T)p(\mathcal{H}_T)}{p(Z_k | \mathcal{H}_F)p(\mathcal{H}_F)} \end{aligned} \quad (5.1)$$

where  $P_T$  is the probability of the true object hypothesis  $\mathcal{H}_T$ ,  $P_F$  is the probability of the false alarm hypothesis  $\mathcal{H}_F$ , and  $Z_k$  is the segment measurements over time.

In practice, it is more convenient to use the log likelihood ratio which is defined as the object score function:

$$\mathbb{O} = \log \frac{P_T}{P_F} \quad (5.2)$$

where  $\mathbb{O}$  is the object score function.

The recursive formula to compute the object score function can be derived as:

$$\begin{aligned}
\mathbb{O}_k &= \log \frac{p(Z_k | \mathcal{H}_T)p(\mathcal{H}_T)}{p(Z_k | \mathcal{H}_F)p(\mathcal{H}_F)} \\
&= \log \frac{p(z_k | \mathcal{H}_T)p(Z_{k-1} | \mathcal{H}_T)p(\mathcal{H}_T)}{p(z_k | \mathcal{H}_F)p(Z_{k-1} | \mathcal{H}_F)p(\mathcal{H}_F)} \\
&= \sum_{i=1}^k l_i^\mathbb{O} + l_0^\mathbb{O}
\end{aligned} \tag{5.3}$$

where

$$l_k^\mathbb{O} = \log \frac{p(z_k | \mathcal{H}_T)}{p(z_k | \mathcal{H}_F)} \tag{5.4}$$

and

$$l_0^\mathbb{O} = \log \frac{p(\mathcal{H}_T)}{p(\mathcal{H}_F)} \tag{5.5}$$

Note that  $z_k$  is not the raw measurement points described in Chapter 3. Here  $z_k$  is information about segments, which can be contributed from different forms of information. In our applications,  $z_k$  consist of the geometric information  $z_k^G$  and the kinematic information  $z_k^K$ . Assuming these contributions are independent,  $z_k$  can be computed as:

$$z_k = z_k^G + z_k^K \tag{5.6}$$

and Equation 5.4 can be rewritten as:

$$\begin{aligned}
l_k^\mathbb{O} &= \log \frac{p(z_k^G | \mathcal{H}_T)}{p(z_k^G | \mathcal{H}_F)} + \log \frac{p(z_k^K | \mathcal{H}_T)}{p(z_k^K | \mathcal{H}_F)} \\
&= l_k^G + l_k^K
\end{aligned} \tag{5.7}$$

where  $l_k^K$  is the object score contributed from kinematic information and  $l_k^G$  is the object score contributed from geometric information. Before describing the formulas to compute  $l_k^K$  and  $l_k^G$ , a technique for eliminating unlikely segment-to-object pairings is introduced next.

### Gating

Gating is a statistical test for connecting likely observation-to-object pairs. Let  $\mu_m$  and  $\Sigma_m$  be the predicted mean and covariance of the object  $m$  in the robot frame and the measurement returned from the perception sensor be  $z$ . The norm of the residual vector can be computed by a Mahalanobis metric:

$$d^2 = (z - \mu_m)^T \Sigma_m^{-1} (z - \mu_m) \tag{5.8}$$

A gate  $G$  can be defined for associating the measurement and the object. A measurement-object correspondence is established when

$$d^2 \leq G \quad (5.9)$$

Figure 5.1 shows an example of gating. If Euclidean distance is used for gating,  $z_k^1$  is chosen instead of  $z_k^2$ . Gating using Mahalanobis distance selects  $z_k^2$  instead of  $z_k^1$  which is more appropriate in a statistical sense.

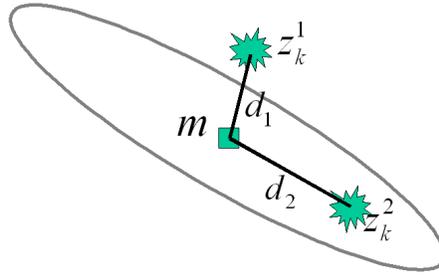


Figure 5.1. Gating.  $z_k^2$  is chosen instead of  $z_k^1$  by using a Mahalanobis metric.

In practice, the gate  $G$  should be chosen to be large enough to compensate both modelled and unmodelled uncertainty. In cluttered environments, gating is only used to eliminate highly unlikely matches. More precise association will be done using the algorithms for solving data association in the cluttered such as the multiple hypothesis tracking (MHT) algorithm.

### Kinematic Contribution to Data Association

If an object is classified as a stationary object  $m$ , the kinematic contribution of the new measurement  $z_k^K$  to this object  $m$  is simplified to the *spatial* contribution only and Equation 5.8 can be directly used to compute  $l_k^K$ :

$$\begin{aligned} l_k^K &= \log \frac{p(z_k^K | \mathcal{H}_T)}{p(z_k^K | \mathcal{H}_F)} \\ &= \log \frac{e^{-d^2/2} / [(2\pi)^{D/2} \sqrt{|\Sigma_m|}]}{1 - p(z_k^K | \mathcal{H}_T)} \end{aligned} \quad (5.10)$$

where  $d^2$  is the norm of the residual vector,  $D$  is the vector dimension and  $\Sigma_m$  is the covariance matrix of the stationary object  $m$ .

If an object is classified as a moving object  $o$ , kinematics of this moving object  $o$  can be learned or selected with the IMM algorithm. For computing  $d^2$ , we can use the mixed predicted pose  $\mu_o$  and its corresponding distribution  $\Sigma_o$ , or use the predicted poses and the

corresponding distributions from the motion models in the model set for performing more accurate association. This procedure is called *maneuver gating* in the tracking literature.

For an object between move and stationary,  $d^2$  is computed with the move hypothesis and the stop hypothesis respectively. The object scores of the hypotheses can be used for hypothesis selection in move-stop hypothesis tracking.

### Geometric Contribution to Data Association

Measurements from laser scanners contain rich geometric information which can be used to aid data association in the small. By applying the sampling and correlation based range image matching algorithm, the relative transformation of a new segment and a object and the corresponding covariance matrix can be obtained. The covariance matrix quantitatively represents the similarity of the new segment and the object. Therefore, we define  $l_k^G$  as:

$$\begin{aligned} l_k^G &= \log \frac{p(z_k^G | \mathcal{H}_T)}{p(z_k^G | \mathcal{H}_F)} \\ &= \log \frac{1/[(2\pi)^{D/2} \sqrt{|\Sigma_S|}]}{1 - p(z_k^G | \mathcal{H}_T)} \end{aligned} \quad (5.11)$$

where  $\Sigma_S$  is the covariance matrix from the sampling and correlation based range matching algorithm.

### Other Contributions to Data Association

The key to solving data association in the small is to exploit information contained in measurements in order to remove ambiguities quickly. Beside the kinematic and geometric information, other contributions should be included if available. For instance, single-related contributions from reflectance or amplitude of laser measurements and color or texture information from cameras could be included using the same mechanism.

## 5.2. Data Association in the Cluttered

Algorithms of data association in the cluttered aim to resolve ambiguities that can not be removed with techniques of data association in the small. Because the ambiguity can not be removed right away, describing the ambiguity properly is the key to disambiguating the situations correctly when new information is received. Figure 5.2 shows an example where both  $z_k^1$  and  $z_k^2$  are inside the gate of object  $o_k^1$ . Given nearly the same object scores, there is no way to eliminate one of them.

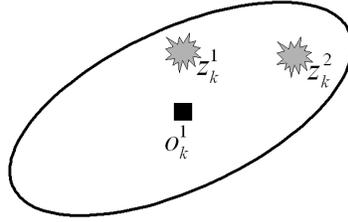


Figure 5.2. Data association in the cluttered.

The data association in the cluttered problem has been studied for several decades. There are a number of algorithms which have been demonstrated successfully in many applications. The multiple hypothesis tracking (MHT) approach (Reid, 1979; Cox and Hingorani, 1996), the joint probabilistic data association (JPDA) approach (Fortmann et al., 1983), and the multidimensional assignment approach (Poore, 1994) are some of the most representative algorithms. The differences and comparisons of these algorithms are beyond the scope of this dissertation and are not addressed here.

Our system applies the MHT method, which maintains a hypothesis tree and can revise its decisions while getting new information. This delayed decision approach is more robust than other approaches. The main disadvantage of the MHT method is its exponential complexity. If the hypothesis tree is too big, it is not feasible to search the whole set of hypotheses to get the most likely set of matching. Note that the MHT algorithm is only applied wherever there is segment-to-object conflict. Figure 5.3 shows the clustering stage of the MHT algorithm where  $o_k^1$ ,  $o_k^2$ ,  $z_k^1$ ,  $z_k^2$ ,  $z_k^3$  belong to the same cluster because of the shared measurement  $z_k^2$ .

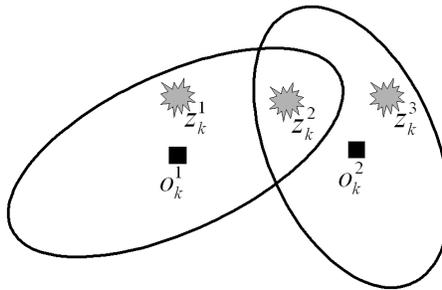


Figure 5.3. Clustering.

Fortunately, with accurate laser measurements, there are usually few association conflicts in our application because of precise data association in the small contributed by kinematic and geometric information.

### 5.3. Data Association in the Large

An example of the data association in the large problem has been illustrated in Section 3.5. When the robot reenters a visited area, loop detection or place recognition has to be done in order to build a globally consistent map (Stewart et al., 2003; Thrun and Liu, 2003; Hähnel et al., 2003).

For data association in the small and in the cluttered, the uncertainty and the ambiguity of the robot and objects' pose estimates can be described well in practice. But for data association in the large, because of accumulated errors and unmodelled uncertainty, the distribution estimates may not describe the uncertainty properly, which means gating can not be performed correctly. Figure 5.4 illustrates the data association in the large problem where the distribution estimates are modelled improperly and can not correctly indicate where the true poses of the robot and objects are. For dealing with this problem, in this section we propose three principles: *covariance increasing*, *information exploiting*, and *ambiguity modelling* where the latter two have been used for solving data association in the small and in the cluttered.

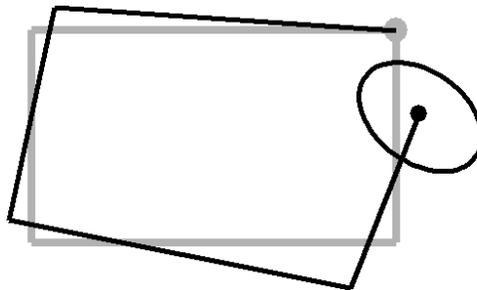


Figure 5.4. Data association in the large.

#### Covariance Increasing

Although the distribution may not describe the uncertainty properly, it still provides useful information for recognizing the current measurement in the built map. In the tracking literature, (Li, 1998) has presented a theoretical conclusion that the covariance matrix from the Kalman filter should be increased for dealing with the missed detection problem. Similarly, instead of searching the whole built map, only the built map within the gate of the increased covariance is verified. Because of the unmodelled uncertainty sources, it may

be difficult to decide how much the covariance matrix should be increased theoretically. In practice, the covariance matrix can be increased in the way of wave propagation until loops are detected. Note that loop detection is only activated whenever there is an inconsistency in the global map. Figure 5.5 illustrates an example of covariance increasing.

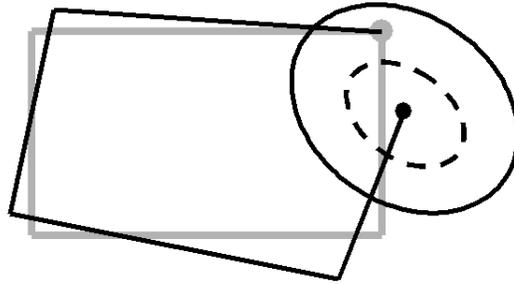


Figure 5.5. Covariance increasing.

### Information Exploiting

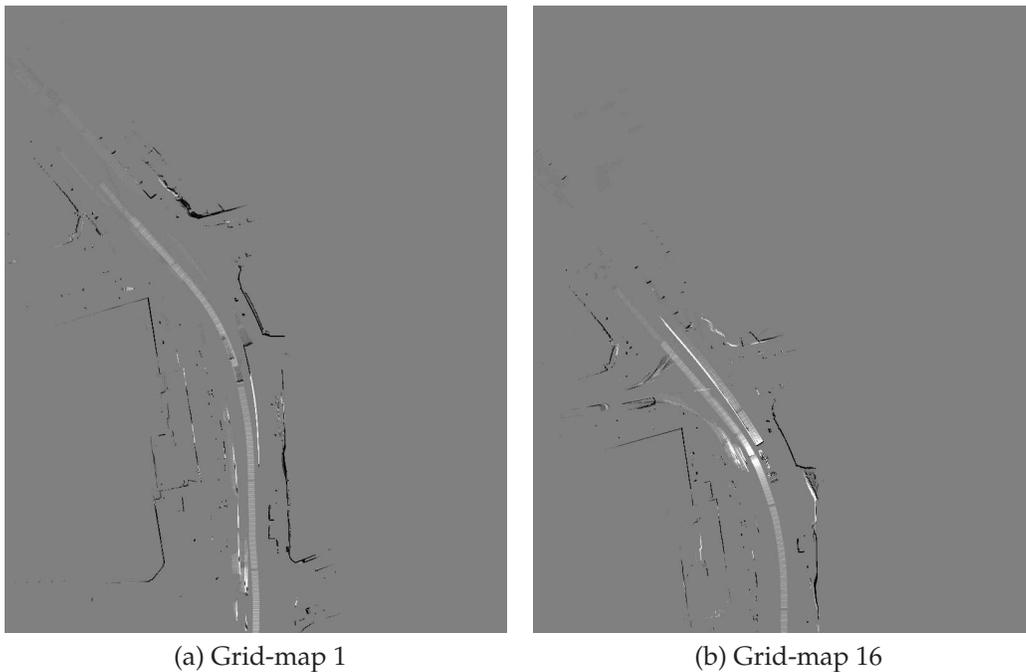
For loop closing, not only *recognizing* but also *localizing* the current measurement within the global map has to be accomplished. As addressed in data association in the small, including geometric information can be greatly beneficial to data association or recognition. Unfortunately, because of *temporary stationary objects*, *occlusion*, and *low object saliency scores*, recognizing and localizing places are difficult even with the proper information about which portions of the built map are more likely.

Because of temporary stationary objects such as cars stopped by traffic lights and parked cars, the currently built stationary object maps may be very different from the global stationary object map. Since the environments are dynamic, stationary objects may be occluded when the robot is surrounded by big moving objects such as buses and trucks. In practice many areas such as bushes and walls do not have high object saliency scores. In these situations, recognition and localization may be incorrect even with the use of geometric information.

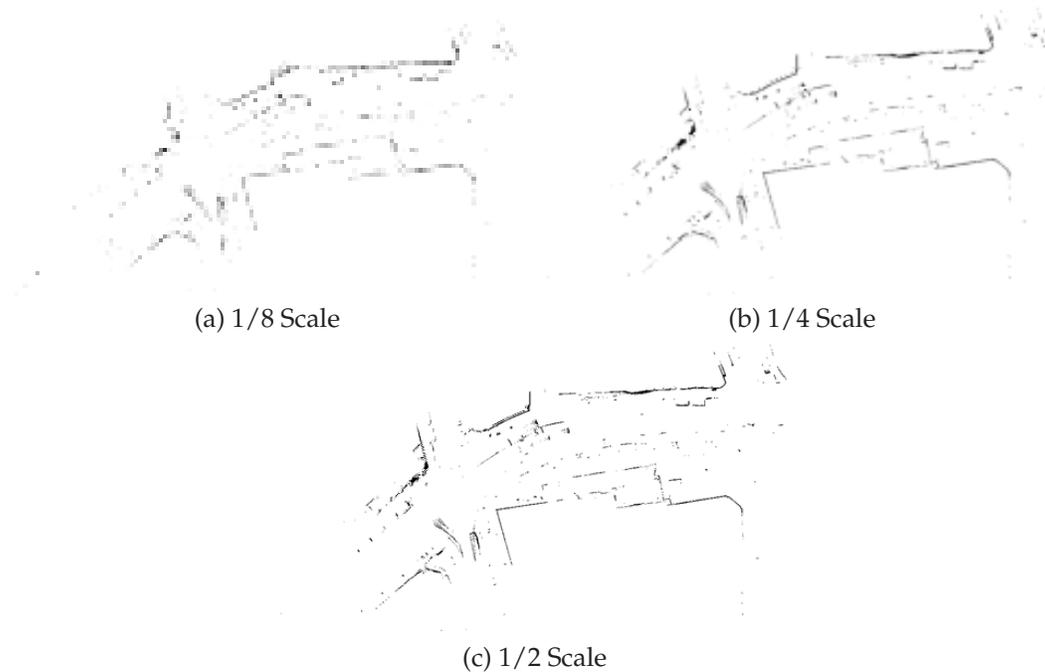
In order to deal with the above situations, big regions are used for loop-detection instead of using raw scans. In large scale regions, large and stable objects such as buildings and street blocks are the dominating factors in the recognition and localization processes, and the effects of temporary stationary objects such as parked cars is minimized. It is also more likely to have higher saliency scores when the size of the regions is larger. In other words, the ambiguity of recognition and localization can be removed more easily and robustly. Because the measurements at different locations over different times are

accumulated and integrated into the local region, the occlusion of stationary objects is reduced as well. Figure 5.6 shows a grid-map pair of the same regions built at different times. Although the details of grid-maps are not the same in the same region because of the described reasons, full grid maps contain enough information for place recognition and localization.

Because grid maps are used, visual image registration algorithms from the computer vision literature can be used for recognition and localization. Following the sampling and correlation based range image matching algorithm, we use the correlation between grid maps to verify the recognition (searching) results, and we perform recognition or searching between two grid maps according to the covariance matrix from the feature-based SLAM process instead of sampling. The search stage is speeded up using multi-scale pyramids. Figure 5.7 shows the recognition and localization results of the examples in Figure 5.6 using different scales.



**Figure 5.6.** The grid-map pair of the same region built at different times: Grid-map 1 and Grid map 16. Different moving object activities at different times, occlusion and temporary stationary objects are shown.



**Figure 5.7.** Recognition and localization results using different scales of grid map 1 and grid map 16. Two grid maps are shown with respect to the same coordinate system.

### Ambiguity Modelling

In cases that information exploiting provides more than one feasible recognition result, the ambiguity should be described properly for a later but firm decision as addressed in data association in the cluttered.

Since the ambiguity in our experiments can be removed quickly and reliably using the described information exploiting based algorithms, we increase the difficulty of the problem by cutting the data set of 21 grid maps into 2 disjoint sequences. We assume that these two sequences are collected from two robots and the relative starting locations of these two robots are unknown. Now the problem is to build a joint map using these two sequences. Figure 5.8 shows these two sequences, one is grid map 1-14 and another is grid map 15-21. Figure 5.9 and Figure 5.10 show the details of these grid maps.

Because the relative starting locations of these two sequences are unknown, recognizing and localizing places have to be performed *globally* in which the saliency score of a grid map may not be good enough to remove the ambiguity. Figure 5.11 shows the bar graph of the maximum correlation values of the grid map pairs between the grid map 1-14 sequence and the grid map 15-21 sequence. Figure 5.12 shows two slices of Figure 5.11 in which multiple possible matches are found in the place recognition of grid map 12 and grid map 13.

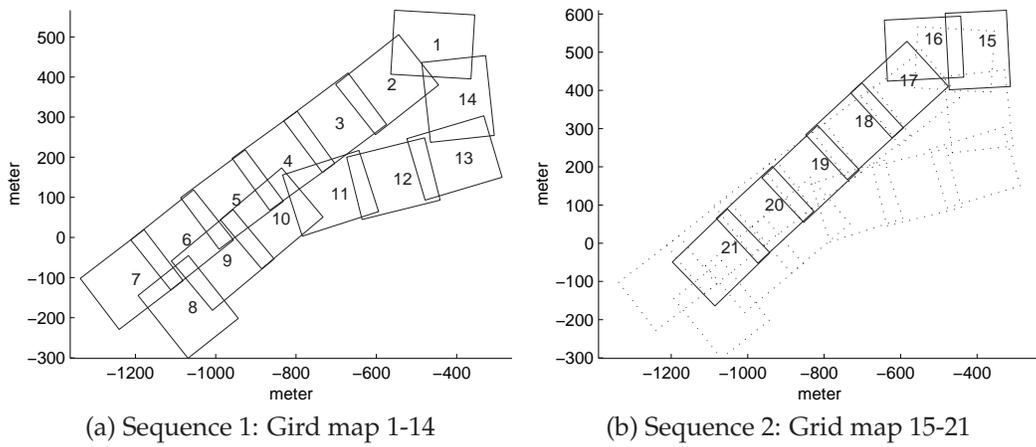


Figure 5.8. Two sequences. The relative starting locations of these two sequences are assumed to be unknown.

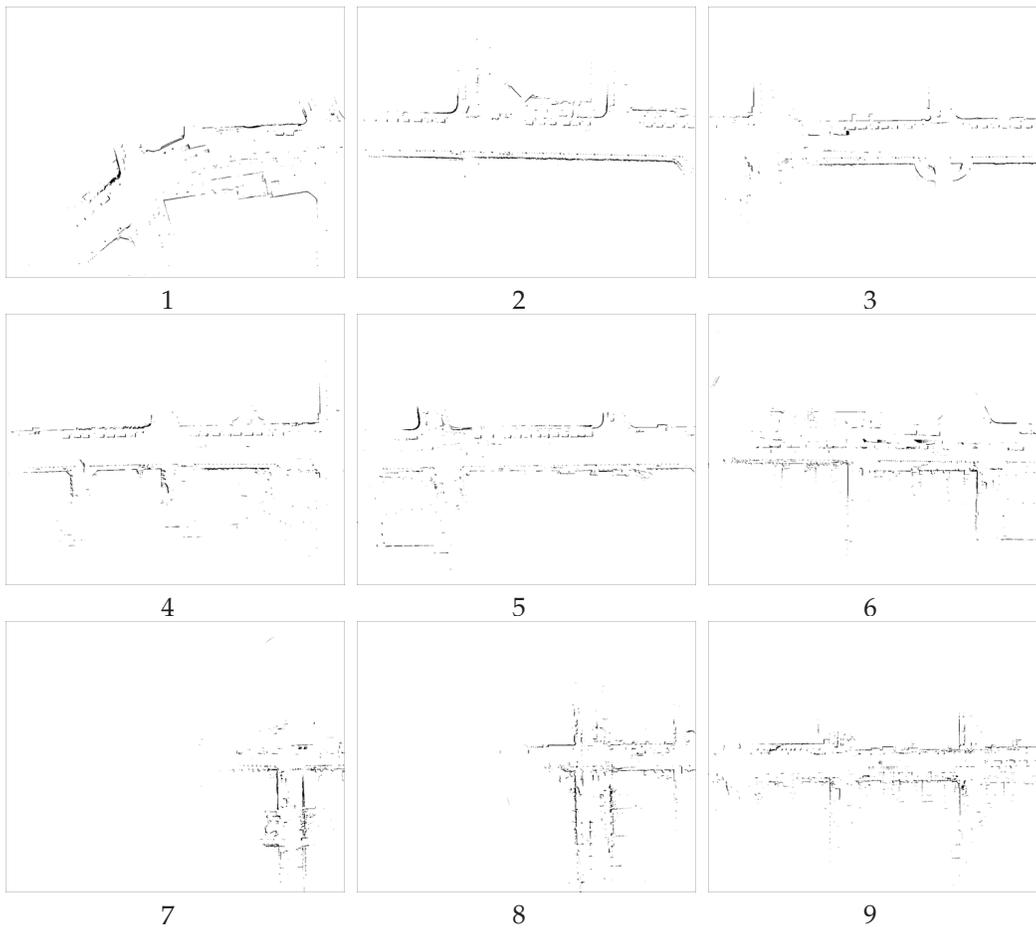


Figure 5.9. Details of grid map 1-9. Measurements associated with moving object are filtered out.

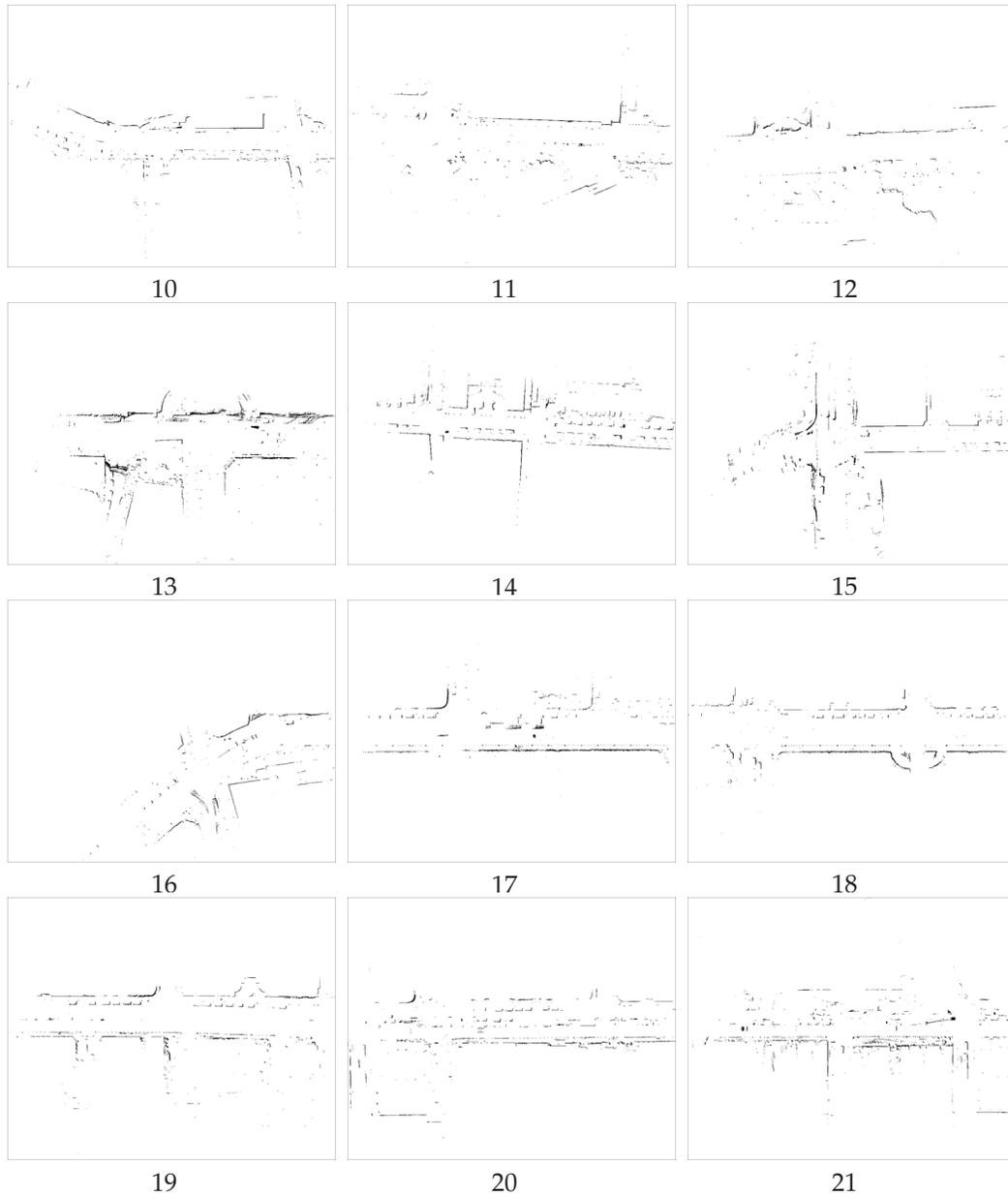


Figure 5.10. Details of grid map 10-21. Measurements associated with moving object are filtered out.

In other words, grid-map 12 and grid-map 13 can not remove the ambiguity. Practically, it can be solved by selecting a larger grid map or using multiple consecutive grid maps to increase the saliency score for removing the ambiguity. Figure 5.13 shows that the ambiguity is reliably removed with the use of multiple consecutive grid maps where hypothesis  $k$  consists of the sequence pair between the grid map sequence  $k-k+5$  and the grid map

sequence 16-21. Hypothesis 1 has the highest total correlation value, which is verified to be the correct recognition by the ground truth.

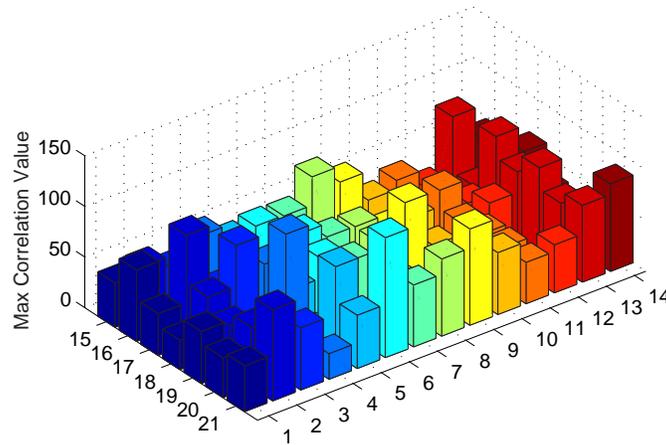


Figure 5.11. The bar graph of the maximum correlation values of the grid map pairs between the grid map 1-14 sequence and the grid map 15-21 sequence using 1/8 scale grid maps.

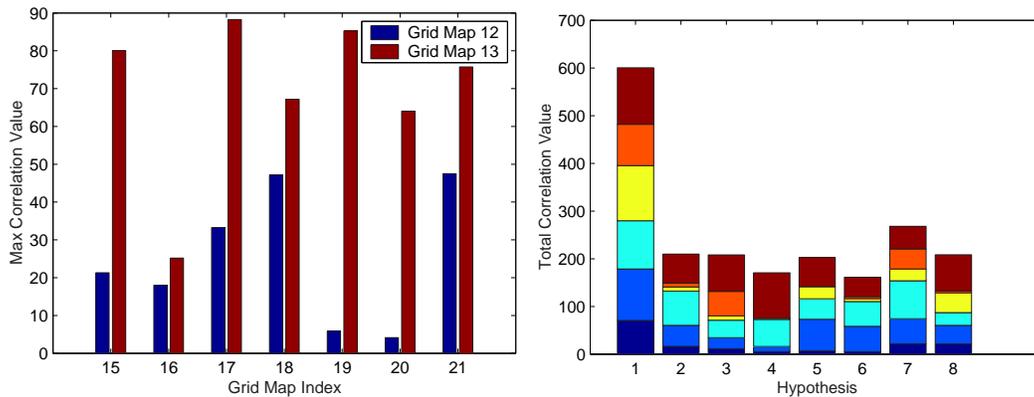


Figure 5.12. The slices of Figure 5.11.

Figure 5.13. The total correlation value of the consecutive grid maps. Each hypothesis consists of five consecutive grid maps.

## 5.4. Summary

In this chapter, we addressed data association in the small, in the cluttered and in the large. These problems were tackled by three principles: information exploiting, ambiguity modelling and covariance increasing. With the rich information contained in the laser scanner measurements, not only kinematic but also geometric information are used for aiding

data association. Following these three principles, we demonstrated that data association in the large can be solved in practice.

Now we have the main components for solving simultaneous localization, mapping and moving object tracking. In the next chapter, we will address the implementation issues, which link all these main components together.

## CHAPTER 6

---

### Implementation

*Always look on the bright side of life.*

*– Monty Python*

**I**N the previous chapters, we have established foundations, perception modelling, motion modelling and data association regarding to solve the simultaneous localization, mapping and moving object tracking problem. In this chapter, we link these individual components together and describe the associated implementation issues from both theoretical and practical points of view.

Compared to the previous chapters, the topics in this chapter are more varied. We begin this chapter with the discussion of solving the whole problem globally or locally. We will describe two practical moving object detection algorithms, consistency based detection and moving object map based detection. We will explain data-driven approaches for tackling non-linearity and non-Gaussianity. We will show ample experimental results to demonstrate the feasibility of simultaneous localization, mapping and moving object tracking from ground vehicles at high speeds in urban areas. Finally, we will also point out the limitations of our system in terms of the 2-D environment assumption and sensor failures.

#### 6.1. Solving the Moving Object Tracking Problem Locally or Globally?

Both the formulations of SLAM with DATMO and SLAM with GO show that the uncertainty of the robot pose estimate from SLAM has to be taken into account for performing moving object tracking because measurements are collected from the robot. This means that more uncertain measurements have to be processed in moving object tracking. Since the goal of SLAM is to build a globally consistent map, the uncertainty of the robot pose estimate is maintained and updated with respect to a global coordinate system. If moving

object tracking is also performed in the global frame, motion modelling and data association will be very difficult. Figure 6.1 illustrates that two measurements are within the gate of a moving object because of performing tracking in the global frame.

Fortunately, *locally* performing moving object detection, data association in the small and in the cluttered, and motion modelling does not violate the formulas of SLAM with DATMO and SLAM with GO, and inference and learning of the whole process can still be done globally. Hence, instead of using the global coordinate system, a temporary global coordinate system is selected for performing moving object tracking. With respect to this temporary global coordinate system, the uncertainties of the pose estimates of the robot and moving objects are maintained reasonably in which detection, data association and motion modelling can be performed reliably. Given the distributions with respect to the global coordinate system, the distributions in the temporary global frame can be obtained with the tail-to-tail operation addressed in Section 2.1. Figure 6.1 and Figure 6.2 illustrate the differences between performing tracking globally and locally. In Figure 6.2, data association is performed correctly in the temporary global frame.

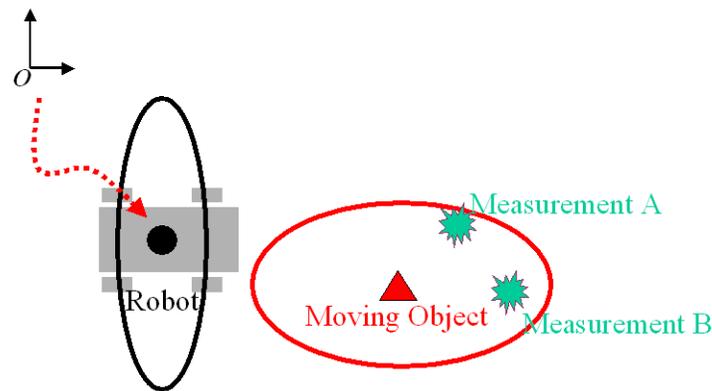


Figure 6.1. Performing detection, data association and motion modelling in a global frame.



Figure 6.2. Performing detection, data association and motion modelling with respect to a temporary global coordinate system.

In practice, once a moving object is detected at the first time, the robot frame could be assigned as the temporary global frame for tracking this moving object. If the hierarchical object based representation is used, the origin of the current grid map could be assigned as the temporary global frame.

## 6.2. Moving Object Detection

Recall that SLAM with DATMO makes the assumption that the measurements can be decomposed into measurements of stationary and moving objects. This means that correctly detecting moving object is essential for successfully implementing SLAM with DATMO.

In the tracking literature, a number of approaches have been proposed for detecting moving objects, which can be classified into two categories: *with* and *without* the use of *thresholding*. (Gish and Mucci, 1987) have proposed an approach that detection and tracking occur simultaneously without using a threshold. This approach is called *track before detect* (TBD) in the tracking literature, although detection and tracking are performed simultaneously. However, the high computational requirements of this approach make the implementation unfeasible. (Arnold et al., 1993) have shown that integrating TBD with the dynamic programming algorithm provides an efficient solution for detection without thresholding, which could be a solution for implementing SLAM with GO practically.

In this section, we describe two approaches for detecting moving objects: a consistency based approach and a motion object map based approach. Although these two approaches work with the use of thresholding, the experimental results using laser scanners are satisfying. In addition, move-stop hypothesis tracking can be used to detect moving objects with estimates below the designed threshold.

### Consistency-based Detection

Intuitively, any inconsistent part between the map and the new measurement should belong to moving objects. In (Wang and Thorpe, 2002), we pointed out that this intuition is not totally correct. There are two cases for detecting moving objects:

**Case 1:** From previous scans or the map, we know some space is not occupied. If we find any object in this space, this object must be moving. In Figure 6.3, object A must be a moving object.

**Case 2:** In Figure 6.4, we can not say that object B is a moving object. Object B may be a new stationary object because object B may have been occluded by object C. What we are sure is that object C is a moving object. Although we can not tell whether or not object B is

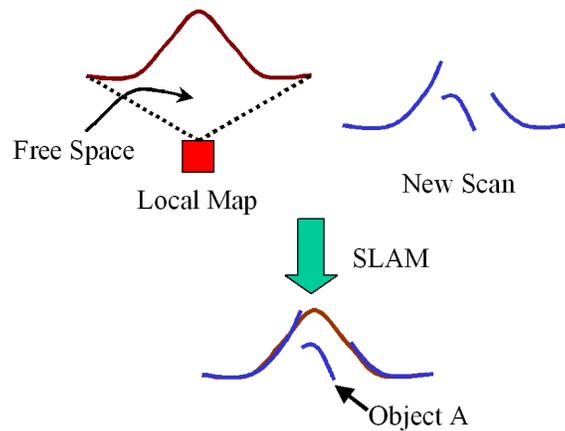


Figure 6.3. Case 1 of detection. See the text for details.

moving by registering only two scans, the previous information does help us to decide the characteristics of object B.

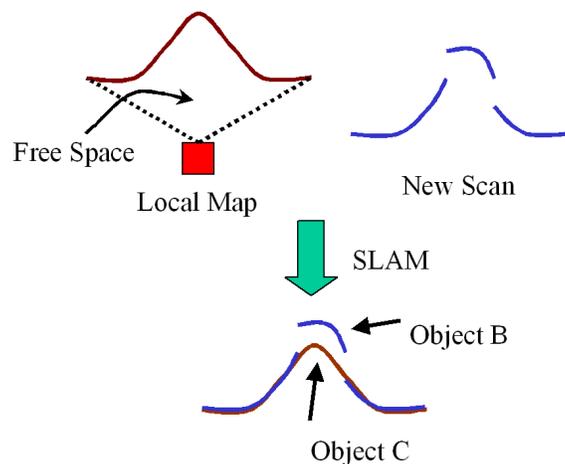


Figure 6.4. Case 2 of detection. See the text for details.

The consistency-based moving object detection algorithm consists of two parts: the first is the detection of moving points; the second is the combination of the results from segmentation and moving point detection for deciding which segments are potential moving objects.

The details are as follows: given a new scan, the local surrounding map, and the pose estimate from SLAM, we first transform the local surrounding map to the coordinate frame of the current laser scanner, and then convert the map from a rectangular coordinate system to a polar coordinate system. Now it is easy to detect moving points by comparing values along the range axis of the polar coordinate system.

A segment is identified as a potential moving object if the ratio of the number of moving points to the number of total points is greater than 0.5. Figure 6.5 shows the results of moving object detection and a red box indicates a moving car recognized by our consistency-based detector.

Note that the consistency-based detector is a motion-based detector in which temporary stationary objects can not be detected. If the time period between consecutive measurements is very short, the motions of moving objects will be too small to be detected. Therefore, in practice an adequate time period should be chosen for maximizing the correctness of the consistency-based detection approach and reducing the load of the more computational expensive move-stop hypothesis tracking algorithm.

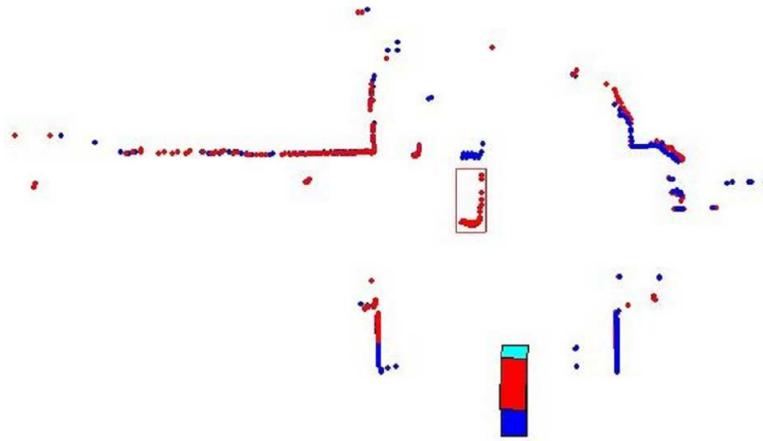


Figure 6.5. Consistency-based detection.

### Moving Object Map based Detection

Detection of pedestrians at very low speeds is difficult but possible by including information from the moving object map. From our experimental data, we found that the data associated with a pedestrian is very small, generally 1-4 points. Also, the motion of a pedestrian can be too slow to be detected by the consistency-based detector. Because the moving object map contains information from previous moving objects, we can say that if a blob is in an area that was previously occupied by moving objects, this object can be recognized as a potential moving object.

### Iterated SLAM with DATMO

Theoretically, making hard decisions such as thresholding leads to an irretrievable loss of information. For SLAM and DATMO, any misclassification of moving objects and

stationary objects could degrade the performance. To successfully implement SLAM with DATMO, detection should be as accurate as possible. By taking advantage of the feasible implementation of SLAM with DATMO, SLAM with DATMO can be performed iteratively until the result converges just as with the ICP algorithm.

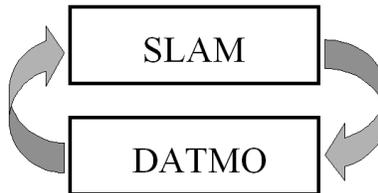


Figure 6.6. Iterated SLAM with DATMO.

### 6.3. Stationary Object Map and Moving Object Map

Instead of discarding information from moving objects, a stationary object map (SO-map) and a moving object map (MO-map) are created to store information from stationary objects and moving objects respectively. Information can easily be accumulated and retrieved by maintaining SO-map and MO-map. Because SO-map only contains stationary object information, SO-map should be clean without any fuzzy areas. In contrast, MO-map is fuzzy because MO-map only contains moving object information. Any inconsistency of SO-map and MO-map provides important information for detecting and correcting mistakes that SLAM with DATMO made. If there are many moving objects passing through an area, any object that appears in this area should be recognized as a moving object as addressed in the previous section.

By integrating information from moving cars and pedestrians, lanes and sidewalks can be recognized. This kind of information is extremely important to robots operating in environments occupied by human beings. In the applications of exploration, robots can go wherever there is no obstacle. However, for tasks in environments shared with human beings, robots have to follow the same rules that people obey. For example, a robot car should be kept in the lane and should not go onto the unoccupied sidewalks. For collision warning, avoidance and planning tasks, velocity estimates of moving objects should not be the only source for deciding if the situation is critical; higher level environment understanding is necessary. The stationary object map and the moving object map provide essential information to accomplish these tasks.

#### 6.4. Data-Driven Approach to Non-Linearity and Non-Gaussianity

Research on SLAM has presented a number of possible methods for solving the SLAM problem such as the extended Kalman filter (Smith and Cheeseman, 1986; Smith et al., 1990), the unscented Kalman filter, the sum-of-Gaussian method (Majumder et al., 2002) and the particle filter (Thrun et al., 2000), which have been proposed for solving the moving object tracking problem in the tracking literature. These methods can be used to solve the simultaneous localization, mapping and moving object tracking problem as well.

The main differences between these methods are the representations of the joint posterior density. In practice, the extended Kalman filter can perform well as long as the estimates are close to the true values and the joint posterior density can be well represented by Gaussian distributions. On the other hand, given enough particles (sample points), the particle filter can provide a complete representation of the joint posterior density, which is the key to dealing with non-linearity and non-Gaussianity. Note that proper perception modelling and motion modelling are still critical to do inference and learning using the particle filter.

Unfortunately, the simultaneous localization, mapping and moving object tracking problem has a much higher dimensionality than the SLAM problem and the moving object tracking problem. Particle filtering may not be feasible because of the curse of dimensionality and history. A few studies such as (Crisan and Doucet, 2002; Verma et al., 2003; Pineau and Thrun, 2002; Roy, 2003) on applying suboptimal approximations to beat the curse of dimensionality and history need to be further studied.

In this dissertation, we apply a *data-driven* approach to tackle the unavoidable non-linearity and non-Gaussianity issues. In *perception modelling*, the sampling and correlation based range matching algorithm are used for analyzing the geometric characteristics of moving objects in the initialization stage of tracking. Once motion models are learned or selected properly, the predicted poses from these motion models are reliable and the ICP algorithm can be used instead of the sampling and correlation based range matching algorithm. In *motion modelling*, move-stop hypothesis tracking is also a data-driven approach. In *data association in the small, in the cluttered and in the large*, new hypotheses are created whenever there are conflicts, and delayed but firm decisions can be obtained, which are data-driven as well.

Our results show that the data-driven approach is a promising methodology for dealing with non-linearity and non-Gaussianity. Some issues such as clustering and model selection need to be studied further.

## 6.5. Experimental Results

Before 2001, experimental data was collected with the Navlab8 vehicle (see Figure 1.7). A SICK PLS100 laser scanner was mounted on the right side of the Navlab8 vehicle, doing horizontal profiling. The preliminary experimental results showed that it is feasible to accomplish localization, mapping and moving object tracking without using measurements from motion sensors. However the algorithms fail when large portions of stationary objects are occluded by moving objects around the Navlab8 vehicle.

Currently, the Navlab11 vehicle (see Figure 1.8) is used to collect data. The Navlab11 vehicle is equipped with motion sensors (IMU, GPS, differential odometry, compass, inclinometer, angular gyro) and perception sensors (video sensors, a light-stripe rangefinder, three SICK single-axis scanning rangefinders). The SICK scanners, one SICK LMS221 and two SICK LMS291, were mounted in various positions on the Navlab11 vehicle, doing horizontal or vertical profiling. The Navlab11 vehicle was driven through the Carnegie Mellon University campus and around nearby streets. The range data were collected at 37.5 Hz with 0.5 degree resolution. The maximum measurement range of the scanners is 81 m. Table 6.1 shows some features of SICK laser scanners. In this section, we show a number of representative results.

Table 6.1. Features of SICK laser scanners. The measurement points are interlaced with 0.25° and 0.5° resolution

SICK Laser Scanner	PLS 100	LMS 221/221/291
Scanning Angle	180°	100°, 180°
Angular Resolution	0.5°, 1°	0.25°, 0.5°, 1°
Maximum Range	~ 51 m	~ 81 m
Collection Rate	6 Hz with 0.5° resolution	37.5 Hz with 0.5° resolution

### Detection and Data Association

Figure 6.7 shows a result of multiple vehicle detection and data association. Five different cars were detected and associated over 11 consecutive scans. This result demonstrates that our detection and data association algorithms are reliable even with moving objects 60 meters away. Additionally, the visual image from the tri-camera system illustrates the difficulties of detection using cameras.

Figure 6.8 and Figure 6.9 show results of pedestrian detection and data association. In Figure 6.8, object 19, 40, and 43 are detected pedestrians, object 17 is a detected car and Object 21 is a false detection. Without using features or appearances, our algorithms detect moving objects based on motion. In Figure 6.9, the visual image shows several stationary

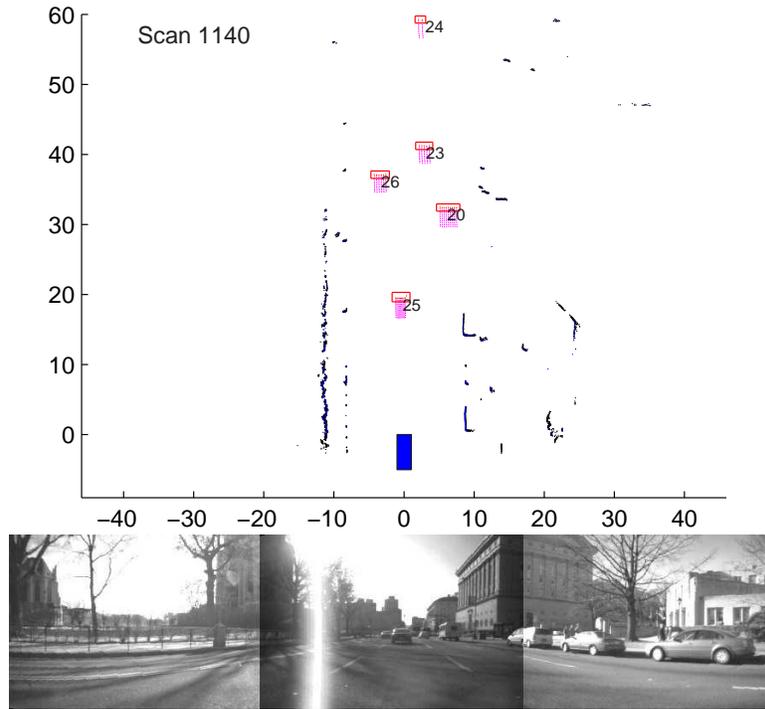


Figure 6.7. Multiple vehicle detection and data association. Rectangles denote the detected moving objects. The segment numbers of the moving objects are shown.

pedestrians that are not detected. Although our approaches cannot classify stationary cars and pedestrians, these temporary stationary objects actually do not have to be dealt with, because their stationary state will not cause any critical threat that the driver/robot has to be aware of, therefore this drawback is tolerable.

Figure 6.10 shows a result of bus detection and data association. Comparatively, Figure 6.11 shows a temporary stationary bus. These big temporary stationary objects have a bad effect upon data association in the large. The approaches for dealing with these temporary stationary objects have been addressed in the previous chapter.

## Tracking

In this section, we show several tracking results of different objects in the real world.

**IMM with the CV and CA models.** Figure 6.12 shows the tracking results of the example in Figure 3.20. The IMM algorithm with the CV and CA models performed well in this case. The distributions of the state estimates described the uncertainty properly.

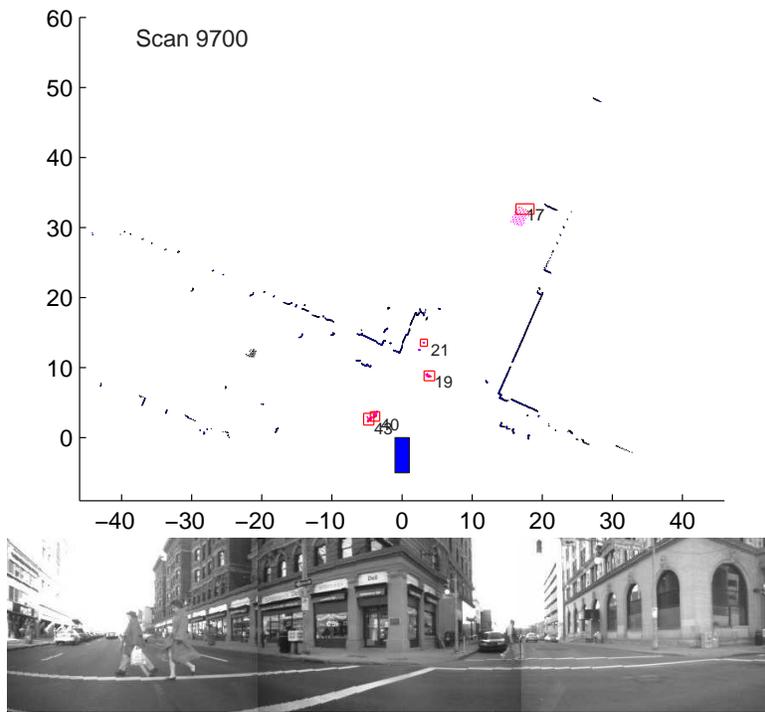


Figure 6.8. Pedestrian detection and data association. See the text for details.

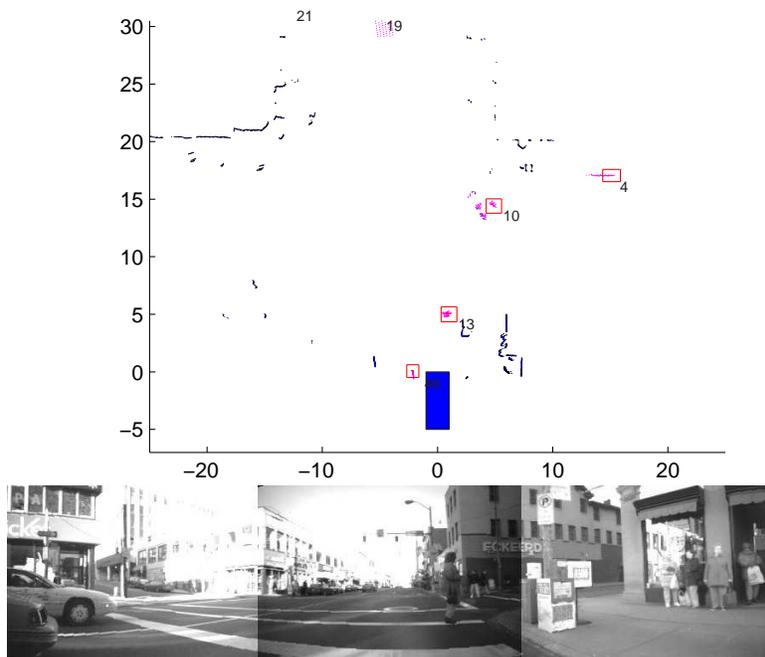


Figure 6.9. Pedestrian detection and data association. The visual image shows several stationary pedestrians, which are not detected by our motion-based detector.

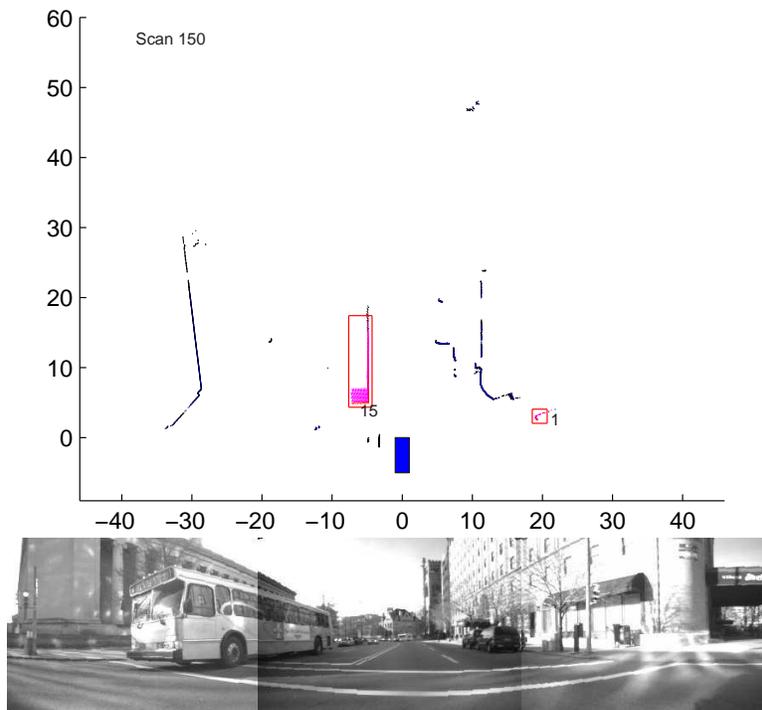


Figure 6.10. Bus detection and data association.

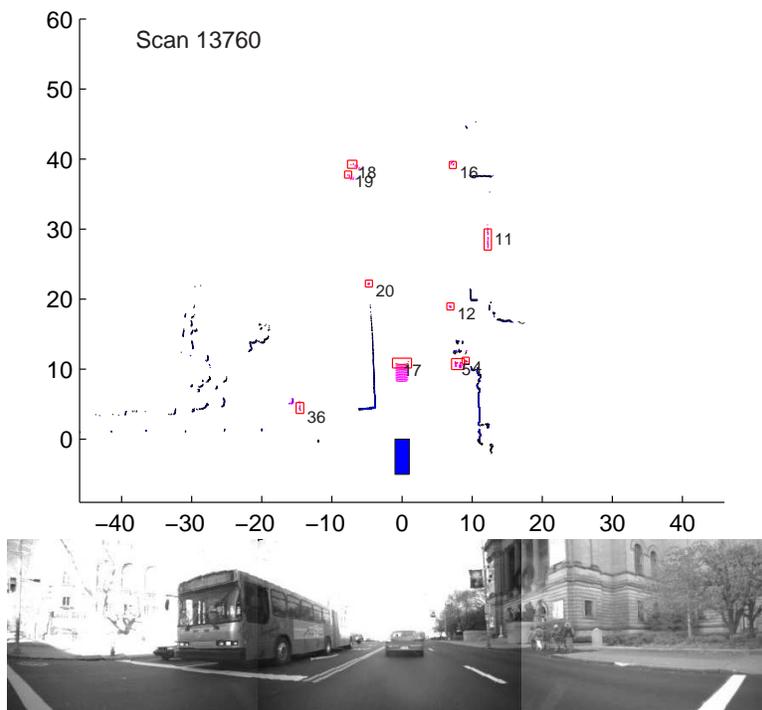


Figure 6.11. Temporary stationary objects. A temporary stationary bus is shown.

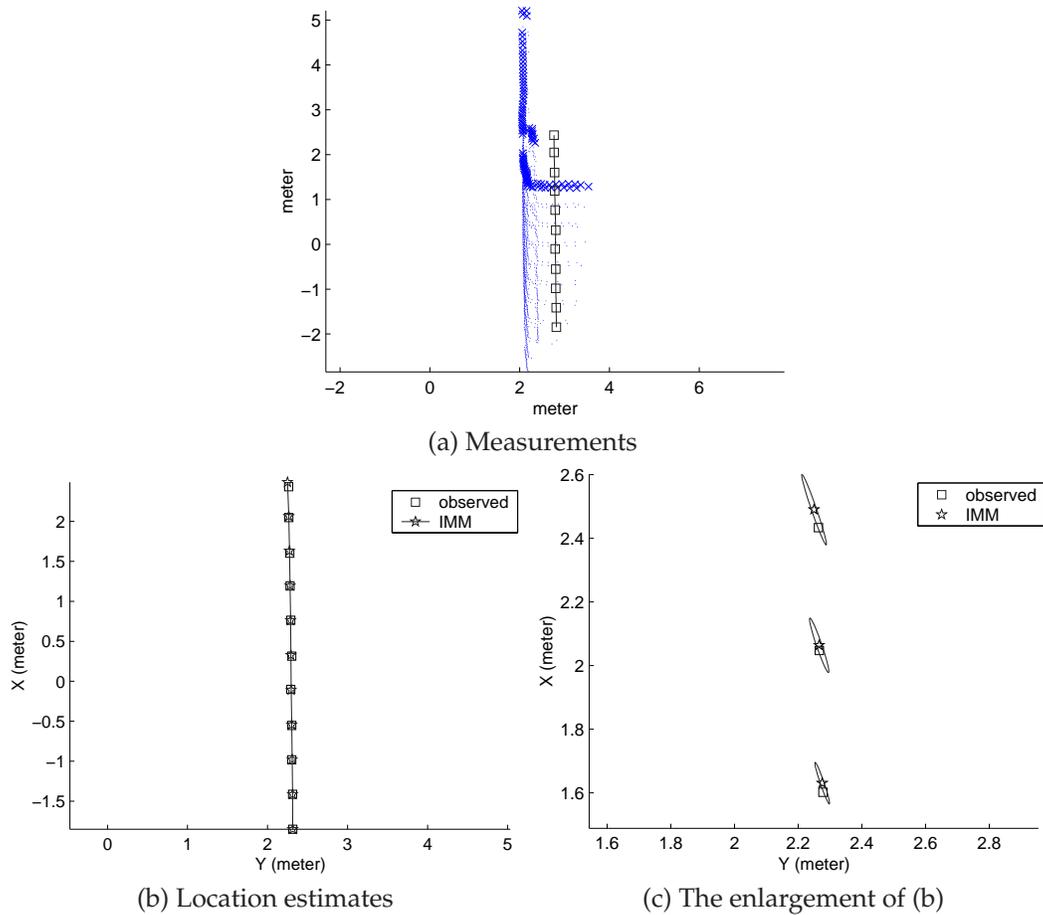


Figure 6.12. Tracking results of the example in Figure 3.20. In (c), the distributions of the state estimates are shown by  $1\sigma$  ellipses.

**Ground Vehicle Tracking.** The previous example showed a very short period tracking in which data association was easy because the tracked object was not occluded. Figure 6.13-6.17 illustrate an example of tracking for about 6 seconds. Figure 6.13 shows the detection and data association results and Figure 6.14 shows the partial image from the tri-camera system. Figure 6.15 shows the raw data of the 201 scans in which object B was occluded during the tracking process. Figure 6.16 shows the tracking results. The occlusion did not affect tracking because the learned motion models provide reliable predictions of the object states. The association was established correctly when object B reappeared in this example. Figure 6.17 shows the speed estimates of these four tracked objects from the IMM algorithm.

**Pedestrian Tracking.** Figure 6.18-6.25 illustrate an example of pedestrian tracking. Figure 6.18 shows the scene in which there are three pedestrians. Figure 6.19 shows the

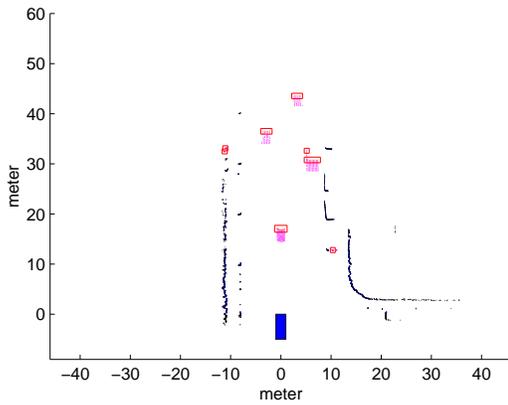


Figure 6.13. Detection and data association results. The solid box denotes the robot.



Figure 6.14. The partial image from the tri-camera system. Four lines indicate the detected vehicles.

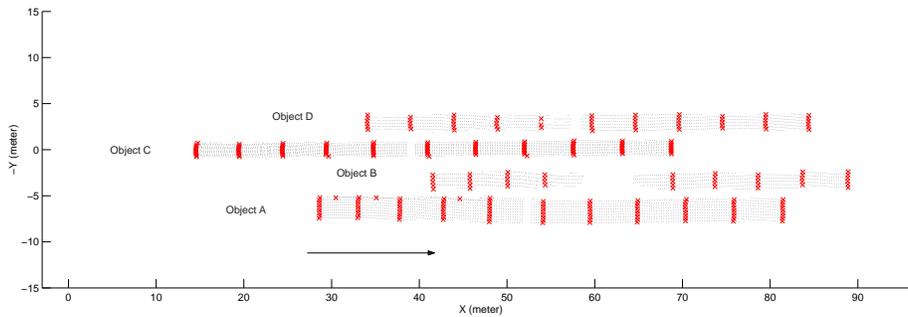


Figure 6.15. Raw data of 201 scans. Measurements associated with stationary objects are filtered out. Measurements are denoted by  $\times$  every 20 scans. Object B was occluded during the tracking process.

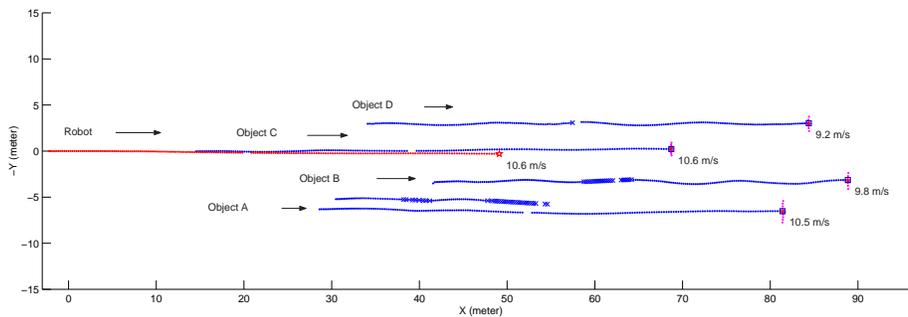


Figure 6.16. Results of multiple ground vehicle tracking. The trajectory of the robot is denoted by the red line and the trajectories of the moving objects are denoted by the blue lines.  $\times$  denotes that the state estimates are from not the update stage but the prediction stage because of occlusion.

visual images from the tri-camera system and Figure 6.20 show the 141 raw scans. Because of the selected distance criterion in segmentation, object B consists of two pedestrians. Figure 6.21 shows the tracking result which demonstrates the ability to deal with occlusion.

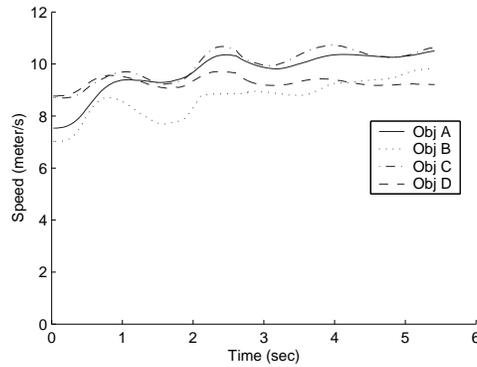


Figure 6.17. Speed estimates.

Figure 6.22 and 6.24 show the speed estimates of object A and B respectively. Figure 6.23 and 6.25 show the probabilities of the CV and CA models of object A and B respectively.

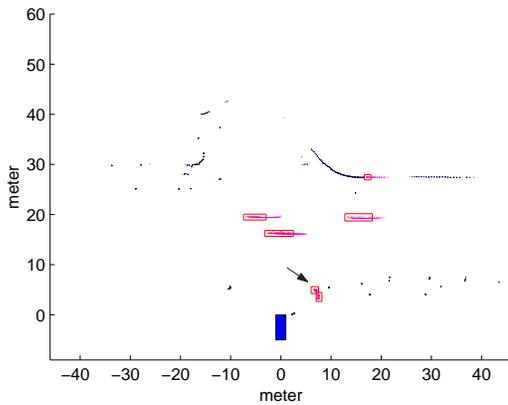


Figure 6.18. An intersection. Pedestrians are pointed out by the arrow.



Figure 6.19. Visual images from the tri-camera system. Block boxes indicate the detected and tracked pedestrians.

**Move-Stop-Move Object Tracking.** Figure 6.26-6.30 illustrate an example of move-stop-move object tracking. Figure 6.26 and Figure 6.28 show the scan from the laser scanner and the visual image from the camera. Figure 6.28 shows the 201 raw scans and the robot trajectory.

Figure 6.29 shows the tracking results using IMM with the CV and CA models and Figure 6.30 shows the speed estimates. As described in Chapter 4, the speed estimates did not converge to zero.

Figure 6.31 shows the result using the move-stop hypothesis tracking algorithm where the stationary motions were identified.

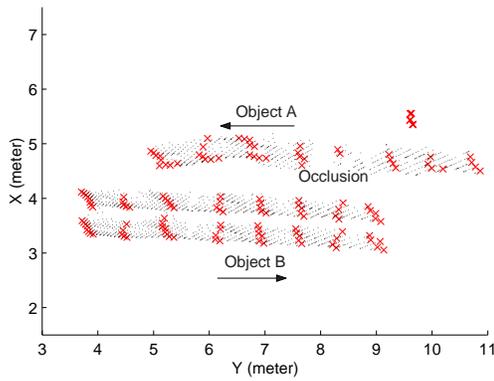


Figure 6.20. Raw data of 201 scans. Measurements are denoted by  $\times$  every 20 scans.

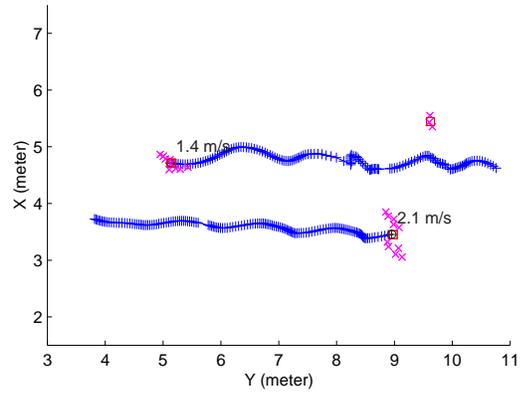


Figure 6.21. Results of multiple pedestrian tracking. The final scan points are denoted by magenta  $\times$  and the estimates are denoted by blue  $+$ .

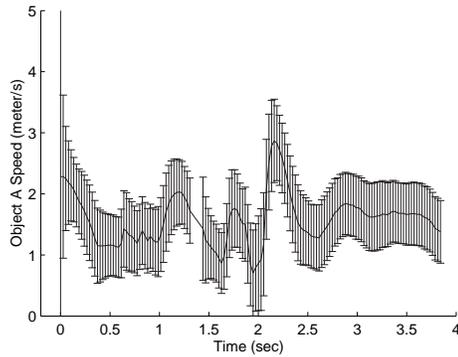


Figure 6.22. Speed estimates of object A.

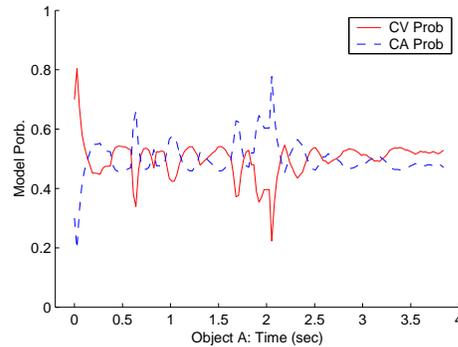


Figure 6.23. Probabilities of the CV and CA models of object A.

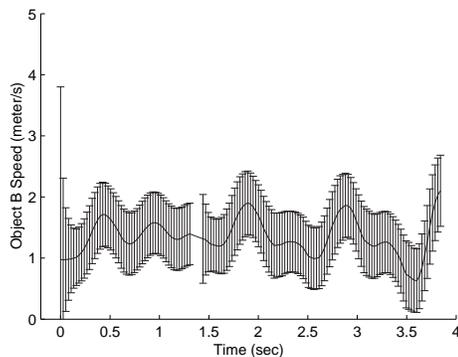


Figure 6.24. Speed estimates of object B.

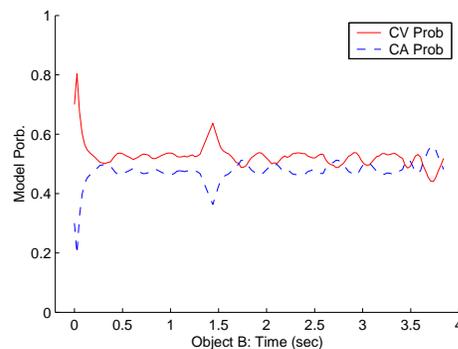


Figure 6.25. Probabilities of the CV and CA models of object B.

### 3D ( $2\frac{1}{2}$ D) City-Sized SLAM

We have demonstrated that it is feasible to accomplish city-sized SLAM in Chapter 3, and Figure 3.28 shows a convincing 2-D map of a very large urban area. In order to

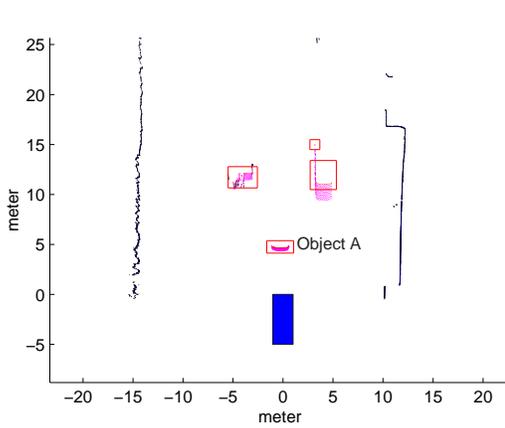


Figure 6.26. The scene.

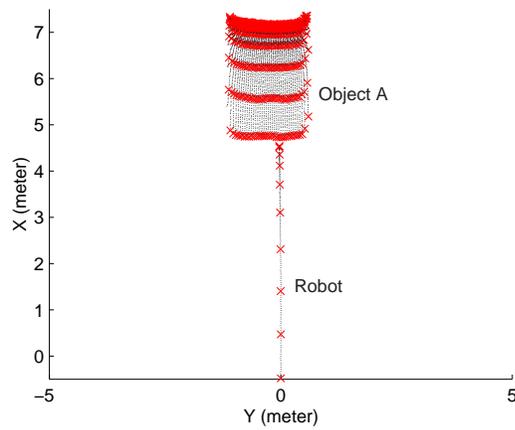


Figure 6.27. 201 raw scans and the robot trajectory. Measurements are denoted by red  $\times$  every 20 scans.

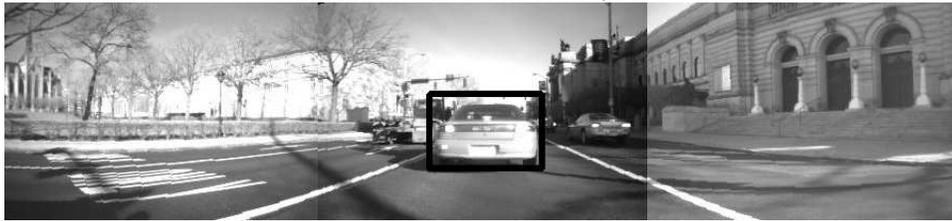


Figure 6.28. The visual image from the tri-camera system. The move-stop object is indicated by a box.

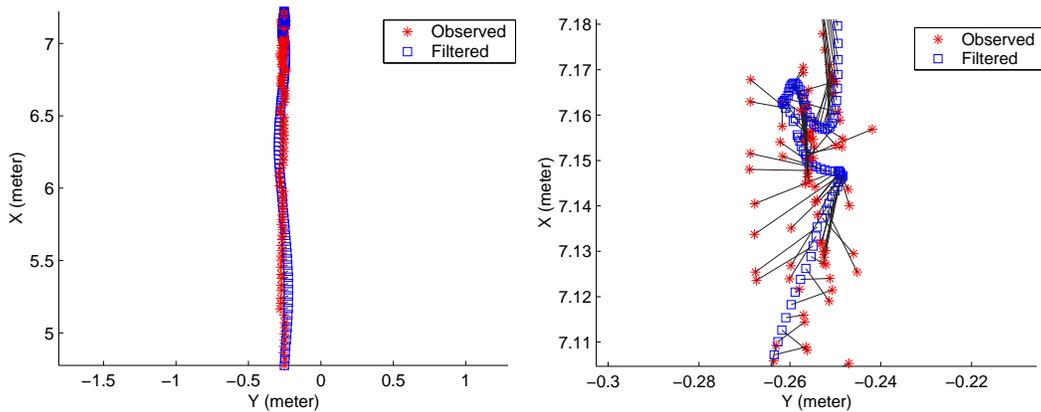


Figure 6.29. The result of move-stop object tracking using IMM with the CV and CA models. On the left: the tracking result. On the right: the enlargement of the left figure. The measurement-estimate pairs are shown by black lines.

build 3-D ( $2\frac{1}{2}$ -D) maps, we mounted another scanner on the top of the Navlab11 vehicle to perform vertical profiling. Accordingly, high quality 3D models can be produced in a minute. Figure 6.32 shows a 3D map of several street blocks. Figure 6.33 shows the 3D

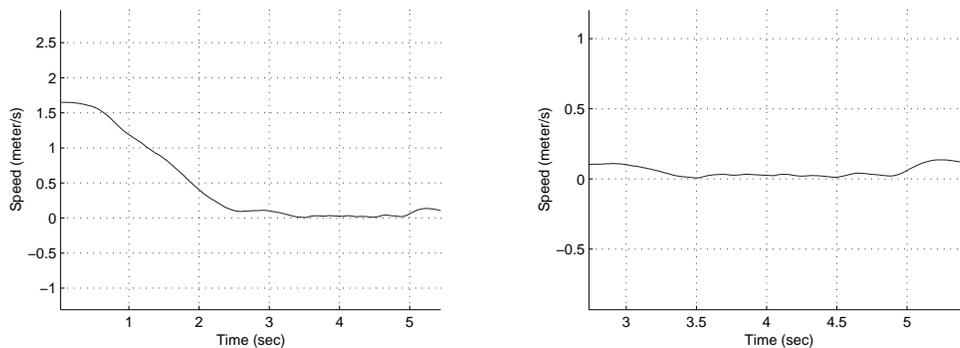


Figure 6.30. Speed estimates from IMM. On the right: the enlargement of the left figure. Note that speed estimates did not converge to zero.

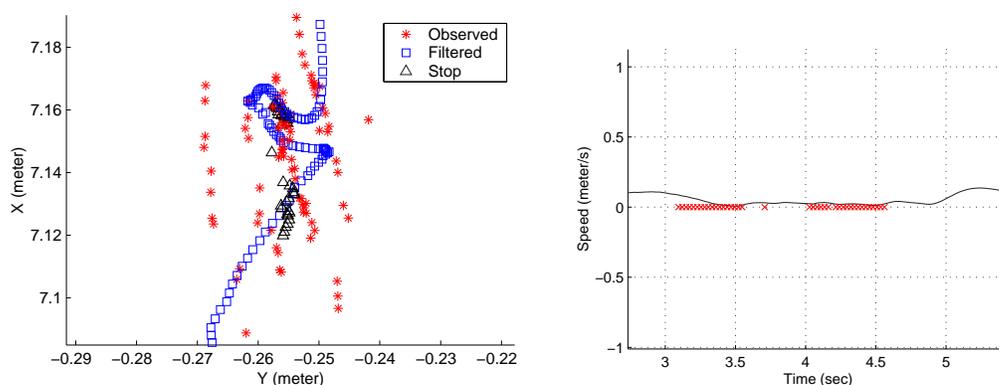


Figure 6.31. The result of tracking using the move-stop hypothesis tracking algorithm. On the left: location estimates. On the right: velocity estimates. Zero velocity estimates are denoted by red  $\times$ .

model of the Carnegie Museum of Natural History. Figure 6.34, Figure 6.35 and Figure 6.36 show the 3-D models of different objects, which may be very useful to applications of civil engineering, architecture, landscape architecture, city planning, etc.

## 6.6. 2-D Environment Assumption in 3-D Environments

Although the formulations derived in Chapter 2 are not restricted to two-dimensional applications, it is more practical and easier to solve the problem in real-time by assuming that the ground is flat. But can algorithms based on the 2-D environment assumption survive in 3-D environments? For most indoor applications, this assumption is fair. But for applications in urban, suburban or highway environments, this assumption is not always valid. False measurements due to this assumption are often observed in our experiments. One is from roll and pitch motions of the robot, which are unavoidable due to turns at high speeds or sudden stops or starts (see Figure 6.37). These motions may cause false

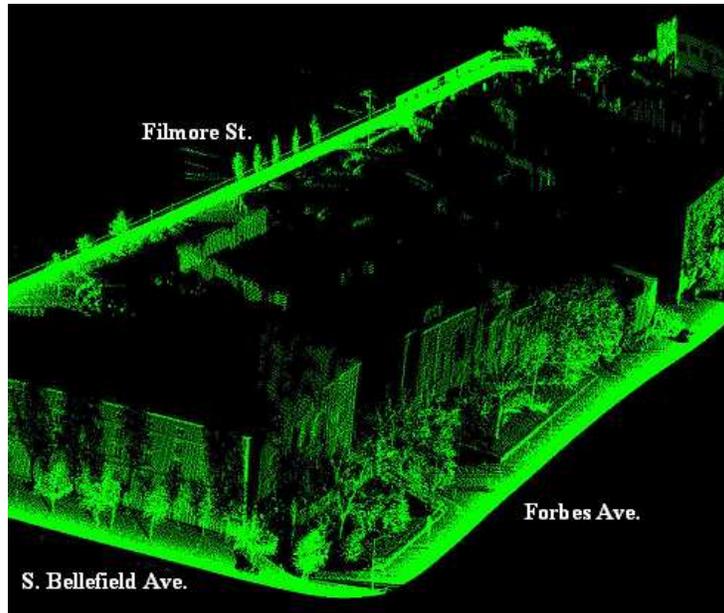


Figure 6.32. A 3-D map of several street blocks.

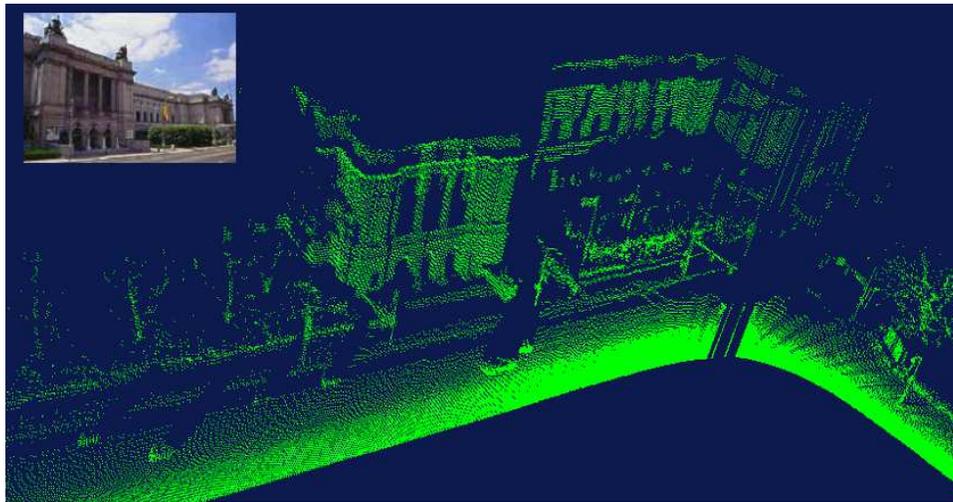


Figure 6.33. A 3-D model of the Carnegie Museum of Natural History.

measurements such as wrong scan data from the ground instead of other objects. Additionally, since the vehicle moves in 3-D environments, uphill environments may cause the laser beam to hit the ground as well (see Figure 6.38). As compared with most metropolitan areas, Pittsburgh has more hills. Table 6.2 shows the steepness grades of some Pittsburgh hills.

In order to accomplish 2-D SLAM with DATMO and SLAM with GO in 3-D environments, it is critical to detect and filter out these false measurements. Our algorithms

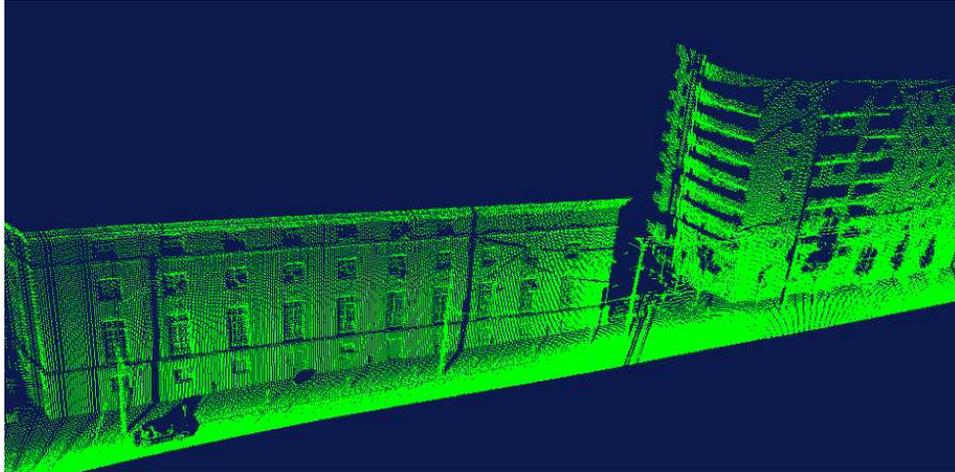


Figure 6.34. 3-D models of buildings on Filmore street.

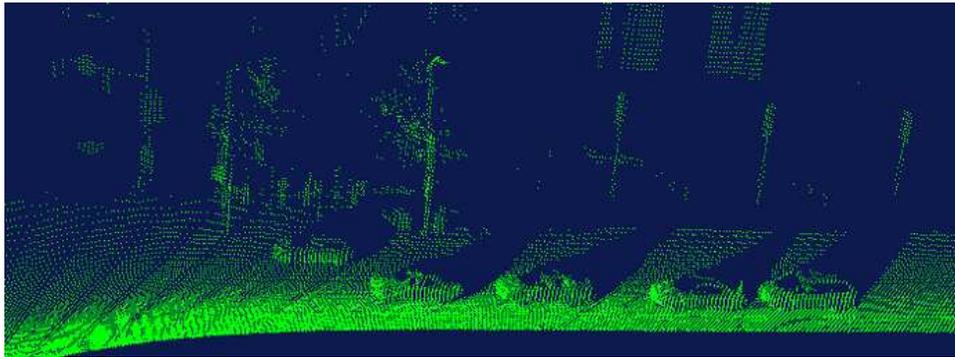


Figure 6.35. 3-D models of parked cars in front of the Carnegie Museum of Art.

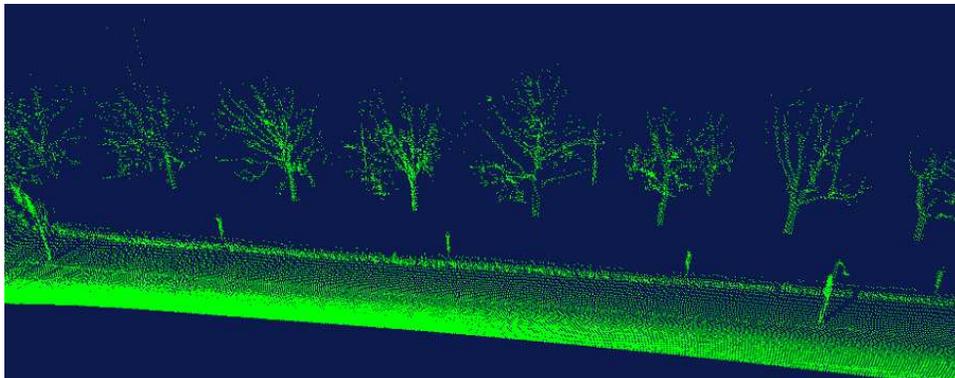


Figure 6.36. 3-D models of trees on S. Bellefield avenue.

can detect these false measurements implicitly without using other pitch and roll measurement. First, the false measurements are detected and initialized as new moving objects by our moving object detector. After data associating and tracking are applied to these

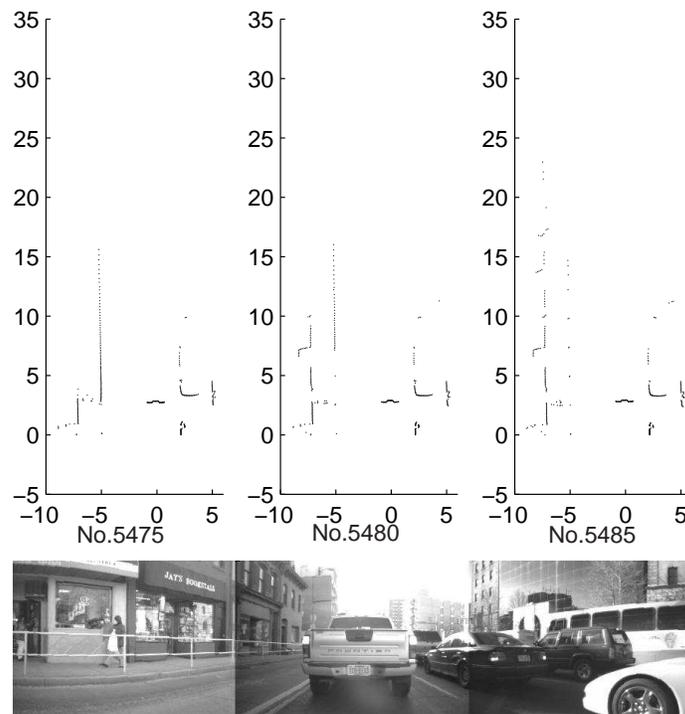


Figure 6.37. Dramatic changes between consecutive scans due to a sudden start.

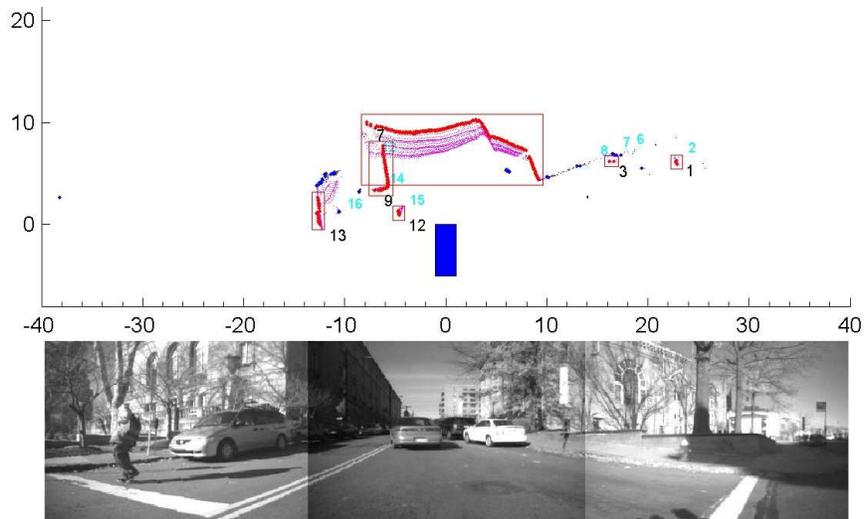


Figure 6.38. False measurements from an uphill environment.

measurements, the shape and motion inconsistency will tell us quickly that these are false measurements. Also these false measurements will disappear immediately once the motion of the vehicle is back to normal. The results using data from Navlab11 show that our 2-D algorithms can survive in urban and suburban environments. However, these big and

**Table 6.2.** Steepness grades of Pittsburgh hills. A 0% grade is perfectly flat and a 100% grade is 45 degrees from the horizontal. This list is from the Pittsburgh Press on Jan 11st, 1987 and was compiled by the Surveys Division and the Snow and Ice Control Program, Dept. of Public Works.

Street	Area	Steepest Grade
Canton Avenue	Beechview	37.00%
Dornbush Street	Homewood	31.98%
Greenleaf Street	Mt. Washington	19.60%
South Negley Avenue	Squirrel Hill	15.81%

fast moving *false alarms* may confuse the warning system and cause a sudden overwhelming fear before these false alarm are filtered out by the SLAM with DATMO or SLAM with GO processes. Using 3-D motion and/or 3-D perception sensors to compensate these effects should be necessary.

## 6.7. Sensor Selection and Limitation

The derived Bayesian formulations for solving the simultaneous localization, mapping and moving object tracking problem are not restricted to any specific sensors. In this section, we discuss the issues on selection and limitations of perception and motion sensors.

**Perception Sensors.** In the tracking literature, there are a number of studies on issues of using different perception sensors (Bar-Shalom and Li, 1995; Blackman and Popoli, 1999). In the SLAM literature, use of different sensors has been proposed as well. For instance, bearing-only sensors such as cameras (Deans, 2002), and range-only sensors such as transceiver-transponders (Kantor and Singh, 2002; Newman and Leonard, 2003) have been used for SLAM.

The fundamentals for using heterogeneous sensors for SLAM, moving object tracking, SLAM with GO, and SLAM with DATMO are the same. The difference is *sensor modelling* according to sensor characteristics. Inference and learning using accurate sensors are more practical and tractable than using imprecise sensors. More computational power and more measurements are needed to extract useful information from imprecise sensors. In applications such as safe driving in urban and suburban environments, robots move at high speeds and have to reason about the surrounding situations as quickly as possible. Therefore, in this dissertation we mainly focus on the issues of using active range sensors.

Although laser scanners are relatively accurate, some failure modes or limitations exist. Laser scanners can not detect some materials such as glass because the laser beam can go through these materials. Laser scanners may not detect black objects because laser light

is absorbed. If the surface of objects is not diffusing enough, the laser beam can be reflected out and not returned to the devices. In our experiments these failure modes are rarely observed but do happen. In Figure 6.39, the measurement from the laser scanner missed two black and/or clean cars, which are shown clearly in the visual image form the tri-camera system. Oppositely, Figure 6.40 shows a failure mode of cameras in which the visual image is saturated by direct sun. But the measurements of the laser scanner are not affected.

Developing better or perfect sensors to resolve these problems may not be feasible in practice. In the next chapter, we will address one of the future extensions of our system, heterogenous sensor fusion, to overcome these limitations.

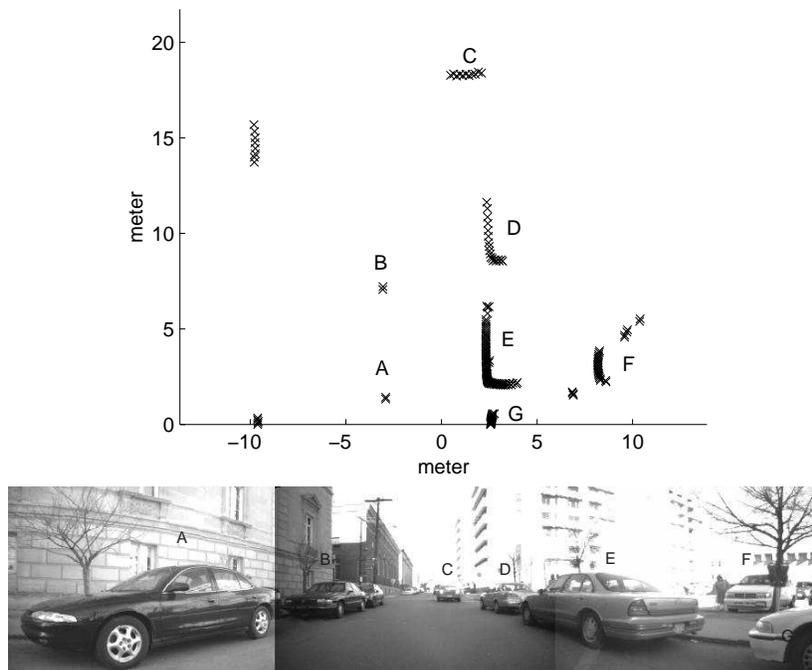


Figure 6.39. The failure mode of the laser scanners. Car A and Car B are not shown completely in the laser scanner measurement.

**Motion Sensors.** In this dissertation, we demonstrate that it is indeed feasible to accomplish simultaneous localization, mapping and moving object tracking using odometry and laser scanners. However, we do not suggest the totally abandonment of inexpensive sensors such as compasses and GPS if they are available. With extra information from these inaccurate but inexpensive sensors, inference and learning can be easier and faster. For instance, for the data association in the large problem, the computational time for searching can be reduced dramatically in the orientation dimension with a rough global orientation estimate from a compass, and in the translation dimensions with a rough global location

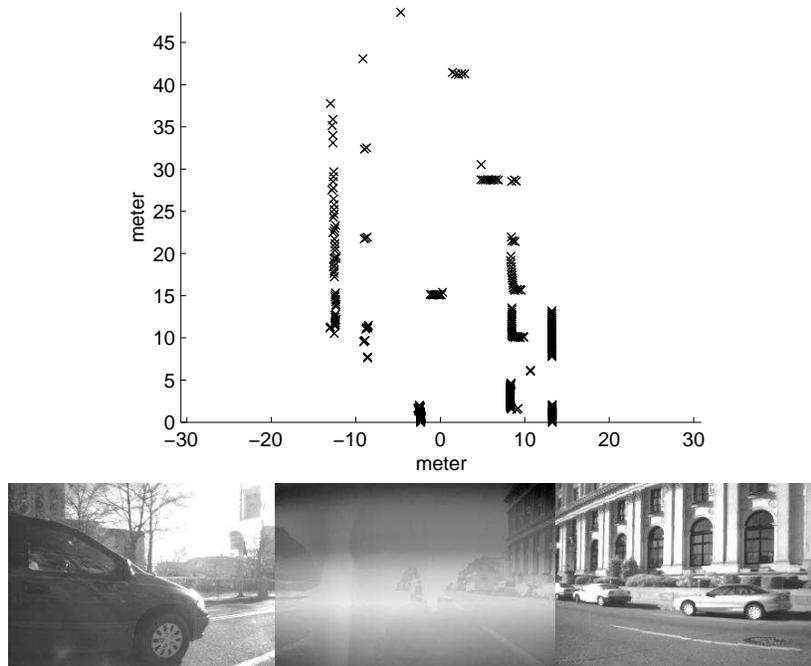


Figure 6.40. The direct sun effect on the regular camera.

estimate from GPS. The saved computational power can be used for other functionalities such as warning and planning.

## 6.8. Ground Truth

It would of course be nice to have ground truth, to measure the quantitative improvement of localization, mapping and moving object tracking with the methods introduced in this thesis. Unfortunately, getting accurate ground truth is difficult, and is beyond the scope of the work in this thesis. Several factors make ground truth difficult:

- Localization: collecting GPS data in city environments is problematic, due to reflections from tall buildings and other corrupting effects.
- Mapping: the accuracy and resolution of the mapping results are better than available digital maps.
- Moving object tracking: any system that works in the presence of uninstrumented moving objects will have a difficult time assessing the accuracy of tracking data.

Some of these difficulties are illustrated by Figures 6.41, 6.42, and 6.43. Figure 6.41 shows the locations of intersections on an available digital map. In Figure 6.43, those same intersections are overlaid on our reconstructed map. In Figure 6.42, the reconstructed map is overlaid on an aerial photo. Qualitatively, the maps line up, and the scale of the

maps is consistent to within the resolution of the digital maps. Quantitative comparisons are much more difficult.

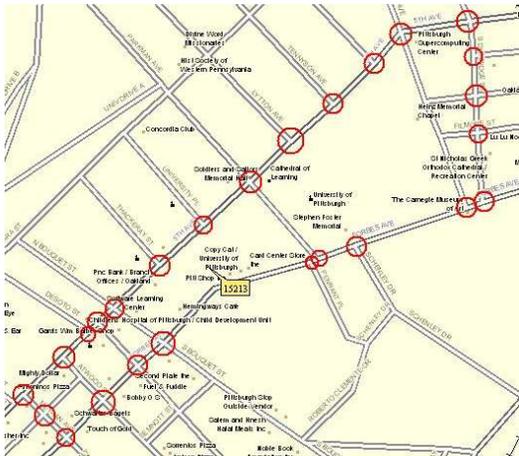


Figure 6.41. An available digital map. The locations of intersections are denoted by circles.



Figure 6.42. The reconstructed map is overlaid on an aerial photo.

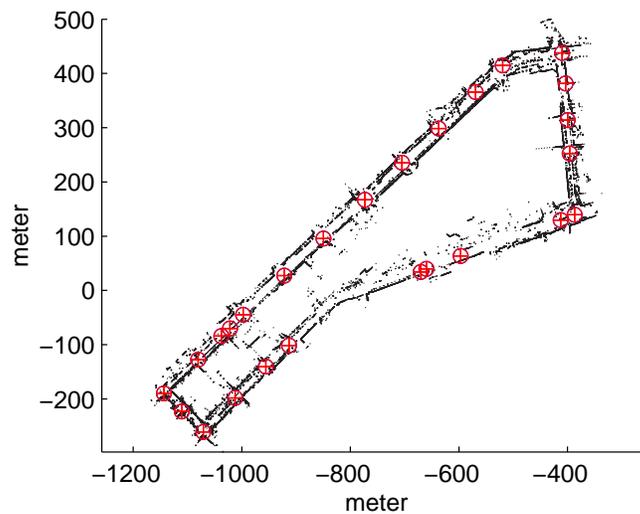


Figure 6.43. The same intersections shown in Figure 6.41 are overlaid on our reconstructed map.

A future project to generate quantitative results would need to:

- characterize the sensors used and their errors.
- carefully characterize the errors of dead reckoning (odometry and heading measurements).
- instrument a few vehicles to be known moving objects, e.g. with accurate GPS or accurate pose estimation systems.

- carefully map a few points on the map to very high resolution, e.g. by using a theodolite to measure distances between corners of a few buildings, or by using carrier phase GPS at the level of the building rooftops, where multipath would not be a factor.

## 6.9. Summary

In this chapter, we have described consistency based detection and moving object map based detection for reliably detecting moving objects using laser scanners. We have shown ample results using data collected from the Navlab11 vehicle, which demonstrate that it is indeed feasible to accomplish simultaneous localization, mapping and moving object tracking from ground vehicles at high speeds in urban areas. Additionally, we pointed out some limitations of our system due to the 2-D environment assumption and sensor failures.



## CHAPTER 7

---

### Conclusion

*The "grand challenge" problem for field robotics is: Create smart, reliable, mobile machines, capable of moving more capably than equivalent manned machines in a wide variety of unstructured environments.*

...

*There are immediate applications, both in terms of complete systems and in terms of robotic components used in other applications. But there is also a rich set of open problems, in fundamental research as well as in applications, that will keep us all busy well into the future.*

*(Thorpe and Durrant-Whyte, 2001)*

**A** new discipline has been established at the intersection of SLAM and moving object tracking in this work. Simultaneous localization, mapping and moving object tracking can be treated as an innovation to seamlessly integrate SLAM and moving object tracking, or an improvement of SLAM and moving object tracking respectively.

In the localization and mapping problem, information associated with stationary objects are positive; moving objects are negative, which degrades the results. Conversely, measurements belonging to moving objects are positive in the moving object tracking problem; stationary objects are negative information, and are filtered out. The central thesis of this work is that both stationary objects and moving objects are positive to the whole problem and they are mutually beneficial.

#### 7.1. Summary

In this dissertation, we established a probabilistic framework for integrating SLAM and moving object tracking. The first solution, SLAM with generic objects, is a general approach which is similar to existing SLAM algorithms but with motion modelling of generic objects. Unfortunately, it has a very high dimensionality and is computationally demanding. In practice, its performance is often degraded because of highly maneuvering objects.

Consequently, we provided the second solution, SLAM with DATMO, in which the estimation problem is decomposed into two separate estimators. By maintaining separate posteriors for stationary objects and moving objects, the resulting estimation problems are much lower dimensional than SLAM with generic objects. This makes it possible to update both filters in real-time. The critical requirement for successful implementation of SLAM with DATMO is correct moving object detection. In addition to move-stop hypothesis tracking, we provided a consistency based approach and a moving object map based approach for detecting moving objects reliably.

Assuming that the static environment assumption is valid, SLAM is still limited to indoor environments, or outdoor environments with specific characteristics. For accomplishing simultaneous localization, mapping and moving object tracking from ground vehicles at high speeds in crowded urban environments, we provided several algorithms and guidelines to eliminate the gaps between theory and implementation. These gaps are categorized into three classes: perception modelling, motion modelling and data association.

We used the hierarchical object based representation to tackle the perception modelling issues of SLAM and moving object tracking. The hierarchical object based representation integrates direct methods, grid-based approaches and feature-based approaches. In addition, we used the sampling and correlation based range image matching algorithm to tackle the uncertain and sparse data issues. Our experimental results have demonstrated that the hierarchical object based representation is an efficient and feasible way to accomplish city-sized SLAM.

Theoretically, motion modelling is as important as perception modelling in Bayesian approaches. Practically, reliable pose predictions from the learned motion models of the robot and moving objects are essential to tasks such as collision warning, dynamic obstacle avoidance and planning. We began with a description of the model selection and model complexity issues. We explained why it is not correct to use the IMM algorithm with the stop model simplified from the constant velocity model for tackling move-stop-move target tracking. The corresponding solutions, the stationary process model and the move-stop hypothesis tracking, are described.

The data association problem is unavoidable because of uncertainty in the real world. We addressed three data association problems in practice: in the small, in the cluttered and in the large. We described three general principles to solve data association: information exploiting, ambiguity modelling and covariance increasing. Geometric information from perception modelling as well as kinematic information from motion modelling are used to

remove the ambiguity. We used the correlation based image registration algorithm along with multi-scale pyramids to solve the revisiting problem robustly and efficiently.

After these theoretical and practical developments, the described formulas and algorithms were carried out with the Navlab8 and Navlab11 vehicles at high speeds in crowded urban and suburban areas. The copious results indicated that simultaneous localization, mapping and moving object is indeed feasible.

## 7.2. Future Extensions

This dissertation raises several interesting topics and there are a number of possible extensions for improving the performances of the system and the algorithms in both theoretical and practical ways.

### Between SLAM with GO and SLAM with DATMO

Since the full solution of simultaneous localization, mapping and moving object, SLAM with GO, is computationally demanding and infeasible in practice, we have presented and implemented the second solution, SLAM with DATMO. The experimental results using laser scanners and odometry have demonstrated the feasibility of SLAM with DATMO. Recall that correct moving object detection is critical for successfully implementing SLAM with DATMO. Nevertheless, in the cases of using sonar and cameras, classifying moving objects and stationary objects may be difficult where a more robust but tractable solution is needed.

Fortunately, it is possible to find an intermediate solution between SLAM with DATMO and SLAM with GO as illustrated in Figure 7.1. In this dissertation, we have pointed out some potential extensions such as detection without thresholding in Section 6.2 and simultaneous multiple moving object tracking in Section 4.7.

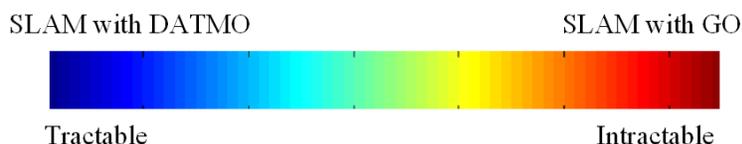


Figure 7.1. Between SLAM with GO and SLAM with DATMO.

### Heterogeneous Sensor Fusion

For understanding complex scenes and increasing reliability and integrity of the robot, heterogeneous sensor fusion is the key. In this work, the Bayesian framework of simultaneous localization, mapping and moving object tracking provides the guidance for fusing

measurements from perception and motion sensors. The experimental results using laser scanners and odometry/IMU are shown to be promising.

Nevertheless, laser scanners may not be sufficient to fully understand a complex scene. For instance, traffic signs, lights and lanes can not be recognized. Besides, laser scanners may fail to produce reliable measurements in the situations addressed in Section 6.7. Therefore, other heterogeneous information should be included and fused to boost reliability and integrity.

Visual images from cameras contain rich information for scene understanding and compensate for some of the disadvantages of laser scanners. There are a number of ways to improve system performance using state-of-the-art algorithms from the computer vision literature. For example, pedestrian detection using laser scanners is difficult because the number of measurement points associated with a pedestrian is often small in our applications. Recognition algorithms can be used to confirm the results of lidar-based detection. Because only portions of the image with high likelihood have to be processed and range measurements from laser scanners can be used to solve the scale issue, the recognition process can be speeded up and run in real-time.

#### 4-D Environments

The real world is indeed four-dimensional, three dimensions for space and one dimension for time. Figure 7.2 shows two examples of 4-D environments. Accomplishing simultaneous localization, mapping and moving objects using 3-D perception and motion sensors is essential to successfully deploy a robot in such environments.



Figure 7.2. 4-D environments.

From a theoretical point of view, the formulation of simultaneous localization, mapping and moving objects in 4-D environments is the same as the described formulas in this

dissertation. However, because of the higher dimensionality in 4-D environments, uncertainty estimation and analysis would be more difficult.

From a practical point of view, perception and motion modelling should be modified according to sensor capability. Because of the richness of 3-D spatial information, data association should be easier and more robust. However, more computational power is required to process large amount of perception and motion data.

### **Toward Scene Understanding**

Estimating the states and motion patterns of the robot and moving objects can be treated as the lowest level of scene understanding. The described algorithms should be sufficient for safe driving in which the robot, or agent, provides proper warnings to assist human drivers. For autonomous driving among human drivers, higher level scene understanding such as *event* or *scenario* recognition is critical.

In the AI literature, there are a number of studies about activity, behavior and interaction modelling. Most related studies are based on simulations or experiments conducted with the use of stationary sensors in indoor or controlled outdoor environments. Our work would make it feasible to conduct experiments in outdoor, dynamic, uncontrolled and very large scale environments. Integrating activity, behavior and interaction modelling into the current framework would lead to a higher level scene understanding.

### **7.3. Conclusion**

It is our hope that this dissertation demonstrates that performing SLAM and moving object tracking concurrently is superior to doing just one or the other. We have answered some important and fundamental questions about formulation, perception modelling, motion modelling and data association. Additionally, we have demonstrated that simultaneous localization, mapping and moving object tracking is indeed feasible from ground vehicles at high speeds in urban environments. We hope that this thesis will serve as a basis for pursuing the questions in fundamental research as well as in applications related to scene understanding or other domains.



## APPENDIX A

---

### Notations and Acronyms

#### A.1. Notations

##### Uncertain Spatial Relationships.

$\oplus(.,.)$	compounding operation
$\ominus(.)$	inverse operation
$\nabla_{\oplus}$	Jacobian of the compounding operation
$\nabla_{\ominus}$	Jacobian of the inverse operation
$\mu$	mean
$\Sigma$	covariance

##### Robot, Stationary Objects and Moving Objects.

$x_k$	true robot state (sensor platform state) at time $k$
$m_k^i$	true location of the stationary object (landmark) $i$ at time $k$
$o_k^i$	true state of the moving object $i$ at time $k$
$y_k^i$	true state of the generic object $i$ at time $k$
$s_k^i$	true motion mode of the generic object $i$ at time $k$
$\mathbf{y}_k^i = \{y_k^i, s_k^i\}$	hybrid state of the the generic object $i$ at time $k$
$M_k = \{m_k^1, m_k^2, \dots, m_k^l\}$	locations of the stationary objects, of which there are $l$ in the world at time $k$
$O_k = \{o_k^1, o_k^2, \dots, o_k^n\}$	states of the moving objects, of which there are $n$ in the world at time $k$
$\mathbf{Y}_k = \{\mathbf{y}_k^1, \mathbf{y}_k^2, \dots, \mathbf{y}_k^{l+n}\}$	states of the generic objects, of which there are $l + n$ in the world at time $k$

##### Perception and Motion Measurements.

$z_k$	perception measurement at time $k$
$z_k^m$	perception measurement associated with stationary objects at time $k$
$z_k^o$	perception measurement associated with moving objects at time $k$
$Z_k = \{z_0, z_1, \dots, z_k\}$	perception measurements up to time $k$
$u_k$	motion measurement or the control input at time $k$
$U_k = \{u_1, u_2, \dots, u_k\}$	motion measurements up to time $k$

##### Perception Modelling.

$h(\cdot)$	perception model
$T$	relative transformation
$T'$	initial guess of the relative transformation
$\hat{T}$	estimate of the relative transformation
$g^{xy}$	occupancy of a grid cell at $\langle x, y \rangle$
$l^{xy}$	log-odd ratio of a grid cell at $\langle x, y \rangle$
$S$	object saliency score

**Motion Modelling.**

$f(\cdot)$	motion model
$\mathcal{M} = \{\mathcal{M}_j\}_{j=1}^r$	motion model set
$F_k$	transition matrix of the system at time $k$
$G_k$	discrete time gain at time $k$
$\mathbf{v}_k$	discrete time process noise at time $k$
$\Gamma$	noise gain
$Q$	covariance of the process noise

**Data Association.**

$P_T$	probability of true object hypothesis $\mathcal{H}_T$
$P_F$	probability of the false alarm hypothesis $\mathcal{H}_F$
$\textcircled{0}$	object score function
$l_k^K$	object score contributed from kinematic information at time $k$
$l_k^G$	object score contributed from geometric information at time $k$

**A.2. Acronyms**

CA	Constant Acceleration motion model
CV	Constant Velocity motion model
DBN	Dynamic Bayesian Network
EKF	Extended Kalman Filter
GPB	Generalized Pseudo-Bayesian
ICP	Iterated Closest Point
IMM	Interacting Multiple Model filtering method
JPDA	Joint Probabilistic Data Association
MHT	Multiple Hypothesis Tracking
MOT	Moving Object Tracking
MTT	Multiple Target Tracking
MVD	Minimum Detection Velocity
Navlab	Navigation Laboratory
PDA	Probabilistic Data Association
PF	Particle Filter
SCRIM	Sampling and Correlation based Range Image Matching
SLAM	Simultaneous Localization and Mapping
SLAM with DATMO	Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects
SLAM with GO	Simultaneous Localization and Mapping with Generic Objects
SLAMMOT	Simultaneous Localization, Mapping and Moving Object Tracking
TBD	Tracking Before Tracking
VS-IMM	Variable Structure Interacting Multiple Model

## BIBLIOGRAPHY

---

- J. Arnold, S.W. Shaw, and H. Pasternack. Efficient target tracking using dynamic programming. *IEEE Transactions on Aerospace and Electronic Systems*, 29(1):44–56, Jan 1993.
- Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques, and Software*. YBS, Danvers, MA, 1988.
- Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS, Danvers, MA, 1995.
- O. Bengtsson and A.-J. Baerveldt. Localization in changing environments - estimations of a covariance matrix for the IDC algorithm. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.
- P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(2):239–256, 1992.
- Samuel Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, 1999.
- Andrew Blake, Ben North, and Michael Isard. Learning multi-class dynamics. In *Proceedings of Advances in Neural Information Processing Systems*, pages 389–395. MIT press, 1999.
- H. A. P. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE Trans. On Automatic Control*, 33(8), Aug. 1988.
- Michael Bosse, Paul Newman, John Leonard, Martin Soika, Wendelin Feiten, and Seth Teller. An atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003.
- G. Champleboux, S. Lavallee, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3-D object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 83–89, June 1992.
- Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1991.

## BIBLIOGRAPHY

- Chee-Yee Chong, David Garren, and Timothy P. Grayson. Ground target tracking - a historical perspective. In *Proceedings of the IEEE Aerospace Conference*, volume 3, pages 433–448, March 2000.
- 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):125–136, April 2001.
- Henrik I. Christensen, editor. *Lecture Notes: SLAM Summer School 2002*. 2002. URL <http://www.cas.kth.se/SLAM/>.
- Stefano Coraluppi and Craig Carthel. Multiple-hypothesis IMM (MH-IMM) filter for moving and stationary targets. In *Proceedings of the International Conference on Information Fusion*, pages TuB1–18–23, Montréal, QC, Canada, August 2001.
- Stefano Coraluppi, Mark Luetzgen, and Craig Carthel. A hybrid-state estimation algorithm for multi-sensor target tracking. In *Proceedings of the International Conference on Information Fusion*, pages TuB1–18–23, Paris, France, July 2000.
- 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 Trans. on Pattern Analysis and Machine intelligence*, 18(2), Feb. 1996.
- D. Crisan and A. Doucet. A survey of convergence results on particle filtering for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, March 2002.
- Matthew C. Deans. *Bearings-Only Localization and Mapping*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2002.
- Frank Dellaert, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition ( CVPR’99 )*, June 1999.
- Nakju Doh, Howie Choset, and Wan Kyun Chung. Accurate relative localization using odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1665–1670, 2003.
- Hugh Durrant-Whyte. A critical review of the state-of-the-art in autonomous land vehicle systems and technology. Technical Report SAND 2001-3685, Sandia National Laboratories, November 2001.
- A. Elfes. *Occupancy Grids as a Spatial Representation for Mobile Robot Mapping and Navigation*. PhD thesis, Electrical and Computer Engineering/Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1988.
- Alberto Elfes. *Autonomous Robot Vehicles*, chapter Sonar-based real-world mapping and navigation. Springer, Berlin, 1990.

- J. Feldmar and N. Ayache. Rigid and affine registration of smooth surfaces using differential properties. In *Proceedings of the European Conference on Computer Vision*, pages 397–406, Stockholm, Sweden, May 1994.
- R. F. Fitzgerald. Simple tracking filters: Steady state filtering and smoothing performance. *IEEE Transactions on Aerospace and Electric Systems*, 16:860–864, Nov. 1980.
- T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8:173–184, July 1983.
- Thor I. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics AS, 1 edition, 2002. ISBN 82-92356-00-2.
- Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*, July 1999.
- C. Früh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 562–569, Madison, USA, June 2003.
- Michael Garland and Paul Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Visualization '98*, October 1998.
- Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):963–996, 1998.
- H. Gish and R. Mucci. Target state estimation in a multi-target environments. *IEEE Transactions on Aerospace and Electronic Systems*, 23:60–73, Jan 1987.
- J. Guivant, E. Nebot, and S. Baiker. High accuracy navigation using laser range sensors in outdoor applications. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3817–3822, 2000.
- J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map building algorithm for real-time implementation. *IEEE Transaction on Robotics and Automation*, 17:242–257, June 2001.
- J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1999.
- J.-S. Gutmann and C. Schlegel. AMOS: Comparison of scan matching approaches for self-localization in indoor environments. In *Proc. the 1st Euro micro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

## BIBLIOGRAPHY

- Dirk Hähnel, Wolfram Burgard, Ben Wegbreit, and Sebastian Thrun. Towards lazy data association in SLAM. In Raja Chatila and Paolo Dario, editors, *Proceedings of the 11th International Symposium of Robotics Research*, Siena, Italy, October 2003.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.
- Martial Hebert. Active and passive range sensing for robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 102–110, 2000.
- Adam Hoover, Gillian Jean-Baptiste, Xiaoyi Jiang, Patrick J. Flynn, Horst Bunke, Dmitry Goldgof, Kevin Bowyer, David Eggert, Andrew Fitzgibbon, and Robert Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996.
- Andrew Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- Michael I. Jordan. *Graphical Models*. 2003. Forthcoming.
- S. Julier. The scaled unscented transformation, 1999. URL [citeseer.nj.nec.com/julier99scaled.html](http://citeseer.nj.nec.com/julier99scaled.html).
- Simon J. Julier and Hugh F. Durrant-Whyte. On the role of process models in autonomous land vehicle navigation systems. *IEEE Transactions on Robotics and Automation*, 19(1):1–14, February 2003.
- George A. Kantor and Sanjiv Singh. Preliminary results in range-only localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2002.
- Alonzo Kelly. Some useful results for closed-form propagation of error in vehicle odometry. Technical Report CMU-RI-TR-00-20, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 2000.
- Alonzo Kelly and Ranjith Unnikrishnan. Efficient construction of optimal and consistent lidar maps using pose network topology and nonlinear programming. In Raja Chatila and Paolo Dario, editors, *Proceedings of the 11th International Symposium of Robotics Research*, Siena, Italy, October 2003.
- T. Kirubarajan and Y. Bar-Shalom. Tracking evasive move-stop-move targets with an MTI radar using a VS-IMM estimator. In *Proceedings of the SPIE Signal and Data Processing of Small Targets*, volume 4048, pages 236–246, 2000.
- T. Kirubarajan, Y. Bar-Shalom, K.R. Pattipati, I. Kadar, E. Eadan, and B. Abrams. Tracking ground targets with road constraints using an IMM estimator. In *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, March 1998.

- K. Konolige and K. Chou. Markov localization using correlation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999.
- B.-U. Lee, C.-M. Kim, and R.-H. Park. An orientation reliability matrix for the iterative closest point algorithm. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 22(10), Oct. 2002.
- J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D. Koditschek J. Hollerbach, editor, *Proceedings of the Ninth International Symposium on Robotics Research*, 1999.
- John Leonard and Hugh Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1442–1447, 1991.
- X. R. Li. Tracking in clutter with strongest neighbor measurement:i-theoretical analysis. *IEEE Transactions on Automatic Control*, 43(11):1560–1578, Nov 1998.
- Xiao-Rong Li. *Multitarget-Multisensor Tracking: Applications and Advances*, volume 3, chapter 10: Engineer’s guide to variable-structure multiple-model estimation for tracking, pages 499–567. Artech House, Boston, 2000.
- Xiao-Rong Li and Yaakov Bar-Shalom. Mode-set adaptation in multiple-model estimators for hybrid systems. In *Proceedings of the American Control Conference*, pages 1794–1799, Chicago, IL, June 1992.
- Xiao-Rong Li and Yaakov Bar-Shalom. Multiple-model estimation with variable structure. *IEEE Transactions on Automatic Control*, 41(4):478–493, April 1996.
- Feng Lu and Evangelos Miliotis. Robot pose estimation in unknown environments by matching 2d range scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–938, Seattle, WA, June 1994.
- Feng Lu and Evangelos Miliotis. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, March 1997.
- David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- S. Majumder, H. Durrant-Whyte, S. Thrun, and M. de Battista. An approximate bayesian method for simultaneous localization and mapping. Submitted to *IEEE Trans. Robotics and Automation*, 2002.
- Martin C. Martin and Hans Moravec. Robot evidence grids. Technical Report CMU-RI-TR-96-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1996.

## BIBLIOGRAPHY

- Larry Matthies and Alberto Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 727–733, Philadelphia, Pennsylvania, USA, 1988.
- E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1): 103–123, January 1998.
- Donald McLean. *Automatic Flight Control Systems*. Prentice-Hall International (UK) Ltd., 1990.
- C. H. Menq, H. T. Yau, and G. Y. Lai. Automated percision measurement of surface profile in CAD-directed inspection. *IEEE Transaction on Robotics and Automation*, 8(2):268–278, 1992.
- Michael Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.
- Kevin Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, July 2002.
- Paul M. Newman and John J. Leonard. Pure range-only subsea slam. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, September 2003.
- Clark F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, February 2000.
- Mark A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In G. Gottlob and T. Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1157–1164, 2003.
- V. Pavlovic, B. Frey, and T. S. Huang. Time series classification using mixed-state dynamic bayesian networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999.
- X. Pennec and J.-P. Thirion. A framework for uncertainty and validation of 3-D registration methods based on points and frames. *International Journal of Computer Vision*, 1997.
- S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003.
- Joelle Pineau and Sebastian Thrun. An integrated approach to hierarchy and abstraction for pomdps. Technical Report CMU-RI-TR-02-21, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2002.

- A.B. Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Computational Optimization and Applications*, 3:27–57, 1994.
- M. Potmesil. Generating models of solid objects by matching 3d surface segments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1089–1093, Aug 1983.
- D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. On Automatic Control*, 24(6), Dec. 1979.
- S. Rogers. Tracking multiple targets with correlated measurements and maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-24(3):313–315, May 1988.
- Nicholas Roy. *Finding Approximate POMDP solutions Through Belief Compression*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 2003.
- S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithms. In *Proc. the Third Int. Conf. on 3-D Digital Imaging and Modeling*, pages 145–152, May 2001.
- D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1665–1670, 2001.
- Peter J. Shea, Tim Zadra, Dale Klamer, Ellen Frangione, and Rebecca Brouillard. Precision tracking of ground targets. In *Proceedings of the IEEE Aerospace Conference*, volume 3, pages 473–482, March 2000.
- Sick Optics. *LMS 200/211/220/221/291 Laser Measurement Systems Technical Description*, 2003. URL [http://www.sickusa.com/Publish/docroot/Manual\(s\)/LMS Technical Description.pdf](http://www.sickusa.com/Publish/docroot/Manual(s)/LMS%20Technical%20Description.pdf).
- David Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996.
- R. W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, MIL-8:125–139, April 1964.
- Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–58, 1986.
- Randall C. Smith, Matthew Self, and Peter Cheeseman. *Autonomous Robot Vehicles*, chapter Estimating Uncertain Spatial Relationships in Robotics. Springer, Berlin, 1990.
- Benjamin Stewart, Jonathan Ko, Dieter Fox, and Kurt Konolige. The revisiting problem in mobile robot map building: A hierarchical bayesian approach. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, Siena, Italy, 2003.

## BIBLIOGRAPHY

- A. J. Stoddart, S. Lemke, A. Hilton, and T. Renn. Estimating pose uncertainty for surface registration. In *British Machine Vision Conference*, 1996.
- Chuck Thorpe and Hugh Durrant-Whyte. Field robots. In *the 10th International Symposium of Robotics Research*, Lorne, Victoria, Australia, November 2001. Springer-Verlag.
- S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000.
- S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters: theory and initial results. Technical Report CMU-CS-02-112, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA, September 2002.
- Sebastian Thrun. Robotic mapping: A survey. Technical Report CMU-CS-02-111, School of Computer Science, Carnegie Mellon University, Pittsburgh, February 2002.
- Sebastian Thrun, Dieter Fox, and Wolfram Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 1998.
- Sebastian Thrun and Yufeng Liu. Multi-robot SLAM with sparse extended information filters. In Raja Chatila and Paolo Dario, editors, *Proceedings of the 11th International Symposium of Robotics Research*, Siena, Italy, October 2003.
- J. K. Tugnait. Detection and estimation for abruptly changing systems. *Automatica*, 18: 607–615, 1982.
- G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of SIG-GRAPH*, pages 311–318, Orlando, FL, July 1994.
- Naonori Ueda and Zoubin Ghahramani. Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15:1223–1241, 2002.
- Vandi Verma, Sebastian Thrun, and Reid Simmons. Variable resolution particle filter. In *International Joint Conference on Artificial Intelligence*. AAAI, August 2003.
- E. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of IEEE Symposium 2000 (AS-SPCC)*, Lake Louise, Alberta, Canada, Oct. 2000.
- Chieh-Chih Wang. Modeling and control of autonomous underwater vehicles. Master's thesis, National Taiwan University, July 1996.
- Chieh-Chih Wang and Charles Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 2002.
- Chieh-Chih Wang, Charles Thorpe, and Arne Suppe. Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds. In *Proceedings of the IEEE Intelligent*

- Vehicles Symposium*, Columbus, OH, June 2003a.
- Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003b.
- J. Y. Wong. *Theory of Ground Vehicles*. John Wiley & Sons, Inc., 3rd edition, 2001.
- Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Int. J. of Computer Vision*, 12(2):119–152, 1994.
- H. Zhao and R. Shibasaki. Reconstructing urban 3d model using vehicle-borne laser range scanners. In *Proc. the Third Int. Conf. on 3-D Digital Imaging and Modeling*, pages 349–356, May 2001.

**Document Log:**

Manuscript Version 1 — 21 August, 2003

Typeset by  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$  — 21 April 2004

CHIEH-CHIH WANG

ROBOTICS INSTITUTE, CARNEGIE MELLON UNIVERSITY, 5000 FORBES AVE., PITTSBURGH, PA 15213,  
USA, *Tel.* : (412) 268-1858  
*E-mail address:* bobwang@cs.cmu.edu

Typeset by  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$