Learning the State of the World:
Object-based World Modeling for Mobile-Manipulation Robots

by

Lawson L.S. Wong

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

Author .....
Department of Electrical Engineering and Computer Science
January 15, 2016

Certified by .....
Leslie Pack Kaelbling
Panasonic Professor of Computer Science and Engineering
Thesis Supervisor

Certified by .....
Tomás Lozano-Pérez
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

Learning the State of the World:
Object-based World Modeling for Mobile-Manipulation Robots

by

Lawson L.S. Wong

Submitted to the Department of Electrical Engineering and Computer Science
on January 15, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Mobile-manipulation robots performing service tasks in human-centric indoor environments have long been a dream for developers of autonomous agents. Tasks such as cooking and cleaning typically involve interaction with the environment, hence robots need to know relevant aspects of their spatial surroundings. However, service robots typically have little prior information about their environment, unlike industrial robots in structured environments. Even if this information was given initially, due to the involvement of other agents (e.g., humans adding/moving/removing objects), uncertainty in the complete state of the world is inevitable over time. Additionally, most information about the world is irrelevant to any particular task at hand. Mobile-manipulation robots therefore need to continuously perform the task of *state estimation*, using perceptual information to maintain a representation of the state, and its uncertainty, of task-relevant aspects of the world.

Because indoor tasks frequently require interacting with objects, objects should be given critical emphasis in spatial representations for service robots. Compared to occupancy grids and feature-based maps that have been used traditionally in navigation and mapping, object-based representations are still in their infancy. By definition, mobile-manipulation robots are capable of moving in and interacting with the world. Hence, at the very least, such robots need to know about the physical occupancy of space and potential targets of interaction (i.e., objects). In this thesis, I propose a representation based on objects, their 'semantic' attributes (task-relevant properties such as type and pose), and their geometric realizations in the physical world.

Acknowledgments

Many people intersected with my path on this Ph.D. journey.

I must begin by acknowledging my advisors, Leslie Pack Kaelbling and Tomás Lozano-Pérez. The two of them, of course, have defined my academic growth and research direction. Both have been incredibly patient, supportive, and available – the perfect academic parents.

I will miss their presence in the journey ahead.

I also thank Russ Tedrake, the remaining member of my thesis committee, for his incredible enthusiasm and flexibility. His interest in my work, even after I have long moved on to the next topic, never ceases to surprise me. I enjoyed the runs we had.

Within my peers, special mention must go to Patrick Barragán, who has unbelievable character and energy, and who really is juggling way too many things in life. I am honored to have been an important part of his years in the group, despite his numerous other commitments.

By sitting in the same office, I probably spend more time with the G418 crowd than with most other people. Patrick's shoe calendar was a great way to bond with Ariel Anders and Owen Macindoe. The newcomers, Kenji Kawaguchi and Caelan Garrett, have also been fine replacements.

Whenever I come across nitpicky issues about types and other definitions, I go to Gustavo Goretkin. Unfortunately I do not see Alejandro Perez and Ashwin Deshpande as much.

In 2014, there was a sudden increase in lab members: Zi Wang, Zelda Mariet, Clement Gehring, Beomjoon Kim, and Kenji all joined at the same time. Your coherence as a cohort has livened our office area significantly, and your perseverance in academic and research matters is admirable.

Thanks, together with apologies for suffering through my first few iterations at mentorship, must go to the LIS UROPs I have worked with: Sheena Nie, Thanard 'Mod' Kurutach, Charlotte Zhu. I wish you all success in your future careers, research or otherwise.

Other former members of the group to thank include Kaijen Hsiao for introducing me to LIS back in 2008, Meg Aycinena Lippow for being a delightful first year presence in an otherwise-empty G585 office, Sam Davies for always being there, Jennifer Barry for being the 'big sister', Huan Liu for being a special M.Eng. student, and Jared Glover for numerous philosophical discussions, and for providing the object detector which this dissertation greatly depends on. Special mention also goes to George Konidaris and Chris Amato for offering mentorship and advice. Daniel Roy, David Wingate, and Finale Doshi-Velez also played an important role in introducing me to the Bayesian philosophy and Bayesian nonparametrics models.

On behalf of LIS, I thank Teresa Cataldo for being our administrative lifeline.

Many MIT entities and groups have also inspired and supported me along the way: Music, EECS REFS, LIDS, Brain and Cognitive Science, and Biology. The Hong Kong Student Society also gave me a sense of community in my early years here, and I came to know many friends, including Karen Lee, Amos Tai, Wing-Ho Ko, Albert Chow, Clement Chan, Godine Chan, and Vincent Cheung.

Thank you, MIT, CSAIL, Cambridge, and Boston.

Finally, I am grateful to my parents, who gave me the opportunity to pursue this journey; and to my wife Carman, who I met here at MIT, and who has been together on this journey all along.

Contents

1 Introduction **11**

1.1 Background and Context 12

1.2 Overview of Dissertation 14

1.2.1 Semantic World Modeling from Partial Views (Wong et al., 2015) 14

1.2.2 Semi-Static World Modeling with Dependent Dirichlet Process Mixtures . . . 15

1.2.3 Combining Object and Metric Spatial Information (Wong et al., 2014) 17

2 Technical Background **19**

2.1 Data Association: A Select Review 20

2.1.1 Multiple hypothesis tracking (MHT) 21

2.1.2 Markov-chain Monte Carlo (MCMC) methods 22

2.1.3 Random finite sets (RFS) 24

2.2 Bayesian Nonparametric Clustering 28

2.2.1 Finite mixture models 28

2.2.2 Dirichlet process mixture models (DPMM) 30

2.2.3 Dependent Dirichlet process mixture models (DDPMM) 33

2.3 Data Fusion 36

3 Data Association for Semantic World Modeling from Partial Views **37**

3.1 Related Work 39

3.2 The 1-D Colored-Lights Domain 40

3.3 A Tracking-Based Approach 41

3.4 A Clustering-Based Approach 44

3.5 Incorporating View-Level Information and Constraints 46

3.5.1 DPMM-FullView 48

3.5.2 DPMM-Factored 50

3.6 Application to Object Type-and-Pose Estimation 54

3.6.1 Qualitative results 55

3.6.2 Quantitative comparisons 56

3.7 Discussion 60

3.A Appendix: Derivation of posterior and predictive distributions for cluster parameters 61

4 Object-based World Modeling in Semi-Static Environments with Dependent Dirichlet Process Mixtures **63**
4.1 Problem Definition . . . 64
4.1.1 Observation noise model . . . 65
4.1.2 Additional assumption: Cannot-link constraint (CLC) . . . 66
4.2 A Clustering-Based Approach . . . 66
4.2.1 Inference by forward sampling . . . 67
4.2.2 Application of DDPs to world modeling . . . 68
4.3 Incorporating World Modeling Constraints . . . 70
4.4 Approximate Maximum *a Posteriori* (MAP) inference . . . 72
4.4.1 Iterated conditional modes (ICM) . . . 73
4.4.2 A two-stage inference scheme . . . 73
4.5 Experiments . . . 75
4.5.1 Simulation . . . 75
4.5.2 Using robot data from static scenes . . . 77
4.6 Discussion . . . 79
4.A Appendix: Derivation of posterior and predictive distributions for cluster parameters 80

5 Not Seeing is Also Believing: Combining Object and Metric Spatial Information **85**
5.1 Problem Definition and Solution Strategy . . . 86
5.2 The One-Dimensional, Single-Object Case . . . 88
5.2.1 Formulation . . . 88
5.2.2 Cell occupancy posterior . . . 89
5.2.3 Object location posterior . . . 90
5.2.4 Demonstrations . . . 91
5.3 Generalizing to Arbitrary States . . . 92
5.4 Applications . . . 94
5.4.1 Shape-based object identification . . . 94
5.4.2 Physical non-interpenetration constraints . . . 96
5.4.3 Demonstration on robot . . . 98
5.5 An Experiment in Comparing Against a Particle Filter . . . 100
5.6 A Graphical Analysis . . . 104
5.7 Conclusions and Future Work . . . 107

6 Conclusion **109**
6.1 What can we do now? . . . 109
6.2 How might we go further? . . . 110
6.3 Final Remarks . . . 111

List of Figures

1-1 Illustration of the semantic world modeling problem in a static world. 15
1-2 An illustration of the semantic world modeling problem in a semi-static world. 16
1-3 Graphical model depiction of object and metric spatial information. 17
1-4 Demonstration of combining object and metric spatial information on a PR2 robot. . 18

2-1 Metropolis-Hastings algorithm for data association. 23
2-2 An illustration of the dependent Dirichlet process (DDP) construction by Lin et al. (2010). 34

3-1 Illustration of the semantic world modeling problem in a static world. 38
3-2 Inference algorithms for Dirichlet-process mixture models (DPMMs). 46
3-3 A real-world example demonstrating issues with applying a generic DPMM to world modeling. 47
3-4 A concrete example for illustrating issues with the generic DPMM, and alternative sampling schemes that address these issues. 49
3-5 A modified DPMM Gibbs sampler that considers view-level information (DPMM-FullView), and an efficient factored approximation to the sampler (DPMM-Factored). 50
3-6 A illustration of the DPMM-Factored method being applied on the previous real-world example. 53
3-7 Qualitative results for 3 world modeling approaches in 5 representative scenarios. . . 57
3-8 Examples of object detections on real-world Kinect images. 58
3-9 Views from a scenario where multiple-hypothesis tracking (MHT) performs incorrectly. 58
3-10 Views from a scenario where there is significant occlusion and many missed detections. 58

4-1 An illustration of the semantic world modeling problem in a semi-static world. 65
4-2 Two-stage inference algorithm for DDPMM, using ICM and MCMCDA. 74
4-3 Data and true object states in a simulated domain. 75
4-4 Cluster trajectories in the simulated domain, found using three different algorithms. 76
4-5 Inferring object trajectories in robot vision data from static scenes 78

5-1 Illustration of object and metric spatial information, and their interaction. 86
5-2 A 1-D demonstration of using object information to infer occupancy states. 92

5-3 A 1-D demonstration of using occupancy information to infer object state. 93
5-4 Illustration of a 1-D scenario involving an object with model uncertainty. 95
5-5 Illustration of a 1-D scenario involving multiple objects, with a natural representation of a non-interpenetration constraint. 97
5-6 Demonstration of combining object and metric spatial information on a PR2 robot. . 99
5-7 Factor-and-fuse vs. a particle filter on a simulated domain (part 1). 101
5-8 Factor-and-fuse vs. a particle filter on a simulated domain (part 2). 102
5-9 The generic filtering problem within a single representational layer. 104
5-10 The factored filter: Filter independently, fuse on demand. 104
5-11 If propagate too much information forward, double-counting will occur. 105
5-12 Two viable filtering strategies to avoid double-counting. 105
5-13 Propagating fused information back to factors may also cause double-counting. . . . 106

6-1 A probabilistic model for inferring locations of unseen objects. 110
6-2 A Willow Garage PR2 robot in a typical laboratory environment. 111

List of Tables

3.1 Quantitative results, including average accuracy metrics and computation wall times, for the scenarios shown earlier. 59

Chapter 1

Introduction

Mobile-manipulation robots performing service tasks in human-centric indoor environments have long been a dream for developers of autonomous agents. Tasks such as cooking and cleaning involve interaction with the environment, hence robots need to know about relevant aspects of their spatial surroundings. However, service robots typically have little prior information about their environment, unlike industrial robots in structured environments. Even if this information was given initially, due to the involvement of other agents (e.g., humans adding/moving/removing objects), uncertainty in the complete state of the world is inevitable over time. Additionally, most information about the world is irrelevant to any particular task at hand. Mobile-manipulation robots therefore need to continuously perform the task of *state estimation*, using perceptual information to maintain a representation of the state, and its uncertainty, of task-relevant aspects of the world.

Because indoor tasks frequently require interacting with objects, objects should be given critical emphasis in spatial representations for service robots. Compared to occupancy grids and feature-based maps that have been used traditionally in navigation and mapping, object-based representations are still in their infancy. By definition, mobile-manipulation robots are capable of moving in and interacting with the world. Hence, at the very least, such robots need to know about the physical occupancy of space and potential targets of interaction (i.e., objects). In this dissertation, I propose a representation based on objects, their 'semantic' attributes (task-relevant properties such as type and pose), and their geometric realizations in the physical world.

Objects are challenging to keep track of because there is significant *uncertainty* in their states. Object detection and recognition is still far from solved within classical computer vision, and even less so from a robotic vision standpoint. Objects can also be inherently ambiguous because they have the same values for some, or even all, attributes. Besides detection noise, other agents may manipulate objects as well and change object states without informing robots. Compounded over multitudes of objects (thousands or more) and long temporal horizons (days or longer), the above sources of uncertainty give rise to a large and difficult estimation problem.

1.1 Background and Context

Understanding the mobile robot's spatial environment, by deriving a world model from its sensors, has long been a problem of interest to the robotics community (Crowley, 1985; Burgard and Hebert, 2008). Early work typically focused on using ultrasonic range sensors, tracking low-level linear, planar, and corner features as landmarks in a map (Cox and Leonard, 1994). The field of simultaneous localization and mapping (SLAM) soon took off, producing metric maps for mobile robot navigation, and SLAM modules are currently widely available on all mobile robot platforms (Herbst and Fox, 2013). However, since the vast majority of robots did not have manipulators, their main task was navigation, which in the indoors setting mainly requires knowledge of obstacles such as walls and furniture. In particular, objects were typically thought of as being nuisance entities that should be removed from the map during post-processing (Thrun et al., 2005).

It was clear eventually that in addition to features and occupancy information, human-centric concepts were also necessary in maps since in many applications humans need to interact with the maps. For example, the navigation task of "go to the kitchen, then to Alice's office" requires knowledge of which place in the map is the kitchen, which places are offices, and also who occupies which office. These human-centric concepts were referred to as "semantic knowledge", and thus the field of semantic mapping was born (Kuipers, 2000). Most work in this area built off of the work in SLAM, and were conceived as providing labels for entities in existing metric/topological maps (Galindo et al., 2005). Occasionally, these labels themselves had additional abstractions on top to encode higher-level knowledge, such as the fact that kitchens and offices are rooms. Finding the appropriate labels for metric regions / topological nodes is often formulated as a classification problem (Pronobis et al., 2010). The most common classification task for indoor semantic mapping is to recognize a certain region is a room, and to determine its type (kitchen, office, etc.). Kostavelis and Gasteratos (2015) provides a recent survey on semantic mapping.

One intuitive cue for recognizing places is objects. For example, detecting a computer keyboard in a room indicates that the room type is most likely "office" and not "kitchen". This was first recognized by Ekvall et al. (2007), and was subsequently incorporated in many semantic mapping works (Ranganathan and Dellaert, 2007; Vasudevan et al., 2007; Zender et al., 2008; Nüchter and Hertzberg, 2008; Pronobis and Jensfelt, 2012). In these applications, images (typically 2-D), and potentially the objects found by object detectors applied to the images, acted as the labels for the underlying metric/topological map. As the authors of those and numerous other works have found, objects are a very useful cue for determining the place category, to the point that Ranganathan and Dellaert (2007) suggested we can basically use objects as a basic unit of representation to model indoor places (instead of mid-level geometric or visual features).

However, in the current generation of semantic maps, there is no fundamental *representation* of the objects, in the sense that one could not ask, for example, "How many keyboards are there in the room?". Consider a room where there were actually two keyboards, each of which were detected in five images taken in the room (i.e, ten detections in total). Current semantic maps would take that as strong evidence that the room is of type "office". However, they may interpret the detections as

indicating that there are ten keyboards in the room, or, more typically, make no interpretation at all. The reason for this is that such interpretation is *unnecessary* for navigation and place recognition, as evidenced by recent work involving object information, but without explicit recognition. Anati et al. (2012) showed that object-based robot localization is still possible even if 'soft' heatmaps of local image features are used instead of explicit object poses.

Similarly, the recent success of dense 3-D reconstruction has led to the suggestion that dense surface maps / point clouds are also a viable representation of space (Newcombe et al., 2011; Whelan et al., 2015). In these dense maps, each point in a point cloud / surface element in a surface reconstruction is endowed with a semantic label. The resulting reconstructions are very visually-appealing and have great potential due to their superior resolution. However, they are limited to visual sensors, require smooth frame transitions, and often are computationally intensive to process. Nevertheless, they have great potential as a fine representation of space.

Regardless of the map's representation, a map annotated with object detections does not equate to object-level understanding. In mobile-manipulation tasks, we need to understand the objects themselves. There is a fundamental difference between obtaining ten keyboard detections and reporting "office", versus identifying that there are two keyboards, determining which one to pick up, localizing the target to sufficient accuracy for robust manipulation, and truly understanding the functional properties of a keyboard. I argue that semantic maps today are not sufficient for mobile-manipulation tasks, which require precise knowledge about *object states*, including information that may not be visible (but can be inferred from other object-based knowledge). *I cannot cook with a reconstructed cloud of points labeled "wok"; I cook with a wok.*

Object state estimation and world modeling, the focus of this dissertation, considers the acquisition and maintenance of knowledge beyond the point of individual object detections. Within the space of object-based state estimation tasks, perhaps the most basic one is: what objects did the robot perceive, and where are they located in the world? These two properties (type and pose) are examples of object *attributes* that an estimator should track. Additionally, the geometric shape models of objects are tracked as special attributes, used to determine their physical occupancy and realization in the world, thus providing information about feasible motions.

Although I mostly focus on object type-and-pose estimation, I would like to emphasize that it was only chosen as a concrete and familiar proof-of-concept application. Most of the presented related work is specific to this application, whereas the methods in this dissertation are applicable to other semantic attributes and tasks.

1.2 Overview of Dissertation

To measure object states, we rely on attribute detectors, particularly ones operating on 3-D visual data. Object recognition and pose estimation has received widespread attention from the computer vision and robotics communities. With the recent advances in RGB-D cameras, several systems have been developed to detect object types/instances and their 6-D poses from 3-D point clouds (Rusu et al., 2010; Glover et al., 2011; Lai et al., 2012; Aldoma et al., 2013; Marton et al., 2014). I have used one such detector (Glover and Popovic, 2013) as the black-box attribute detector, but the methods developed in this thesis are agnostic to the detector used.

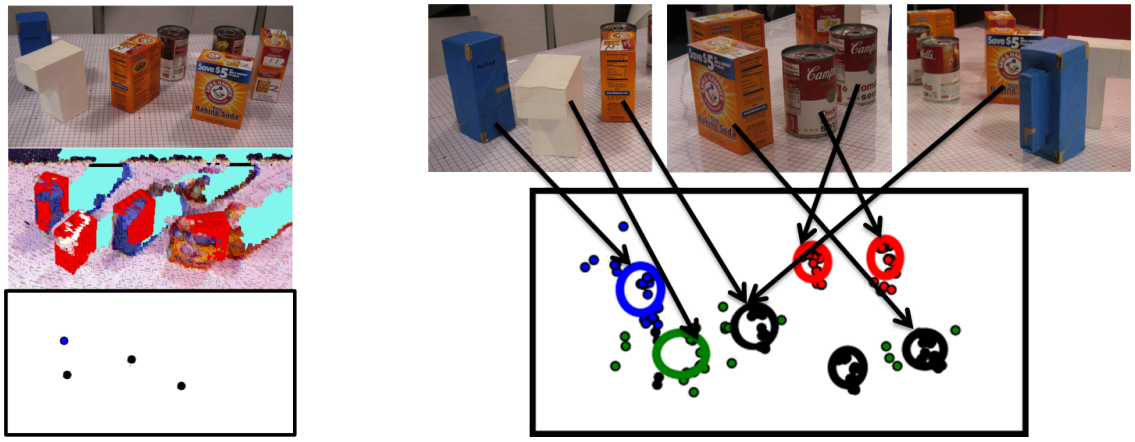
A basic world model could simply use a detector's output on a single image as a representation of the world. However, doing so suffers from many sources of error: sensor measurement noise, object occlusion, and modeling and approximation errors in the detection algorithms. Aggregating measurements across different viewpoints can help reduce estimation error. For example, Hager and Wegbreit (2011) demonstrate the utility of considering a prior 3-D scene model and its potential evolution over scenes. Using this observation as a premise, active perception approaches (e.g., Eidenberger and Scharinger, 2010; Velez et al., 2012; Atanasov et al., 2013) seek the next best view (camera pose) where previously-occluded objects may be visible, typically by formulating the problem as a partially-observable Markov decision process. Because the focus is on planning instead of estimation, active perception is complementary to the world modeling problem, which considers estimation using measurements from an arbitrary collection of camera poses.

The primary challenge in aggregating object detections across multiple views of the world is *identity management*, induced by the fact that measurements often cannot be uniquely mapped to an underlying object. Tackling this data association problem in static scenes forms the first part of my dissertation. From there, I consider how to aggregate detections across time as well, with the added difficulty that the world may change over time. Finally, I consider how to integrate this object-based representation with traditional spatial representations such as occupancy grids, thereby aggregating information across different sensing modalities and representations.

1.2.1 Semantic World Modeling from Partial Views (Wong et al., 2015)

The 'what and where' problem, when considered abstractly on the level of objects and attributes, has a natural generalization: given detections of object attributes only (without knowing which objects generated them), estimate the objects that are present (including their number) and their attributes. I assume the existence of off-the-shelf black-box attribute detectors, such as object recognition and pose estimation modules. Because the information returned from such modules is typically very sparse (at most one detection per object from a single viewpoint), aggregating detections across multiple viewpoints is necessary (see Figure 1-1).

However, this introduces data association issues, because it is unclear which measurements correspond to the same object across different views. I proposed a Bayesian nonparametric batch-clustering approach, inspired by the observation that 'objects' are essentially clusters in joint at-



(a) Single viewpoint **(b)** Aggregation of object detections from multiple viewpoints

Figure 1-1: (a) Given a tabletop scene (top), we want to estimate the types and poses of objects in the scene using a black-box object detector. From a single RGB-D image, however, objects may be occluded or erroneously classified. In the rendered image (middle; detections superimposed in red), three objects are missing due to occlusion, and the bottom two objects have been misidentified. The semantic attributes that result in our representation are very sparse (bottom; dot location is measured 2-D pose, color represents type). A single viewpoint is typically insufficient to identify all objects in a scene correctly.
(b) Aggregation of measurements from many different viewpoints (top) is therefore needed to construct good estimates. However, this introduces data association issues of the type addressed in this work, especially when multiple instances of the same object type are present. From all the object detection data, as shown (bottom) by dots (each dot is one detection), our goal is to estimate the object types and poses in the scene (shown as thick ellipses centered around location estimate; color represents type, ellipse size reflects uncertainty). The estimate above identifies all types correctly with minimal error in pose.

tribute space. Given attribute detections from multiple viewpoints, this algorithm outputs a distribution (in the form of samples) over hypotheses of object states, where a hypothesis consists of a list of objects and (distributions of) their attribute values.

1.2.2 Semi-Static World Modeling with Dependent Dirichlet Process Mixtures

Our operational definition of an "object" is an entity that can be subject to manipulation, and a typical effect of manipulation is some change in the object state (e.g., its pose). Hence the semantic world model for static worlds from the previous section is insufficient. At any given time, however, most objects are not being manipulated (e.g., books on a bookshelf, a home when its occupants are at work). We view indoor environments as being static at most times, changing only at discrete events. An example of this may be a cleaning robot that is turned on every time a home's occupants leave. The state of the world will likely have changed from the previous day, but during the operational period of the robot, the world is essentially static (possibly except for the robot's own manipulation actions, which it can track more easily). We denote such environments as being "semi-static". Figure 1-2 illustrates the full semantic world modeling problem for semi-static environments, allowing for objects to change state over time (epochs).

The way we have posed the data association problem is reminiscent of multiple object tracking

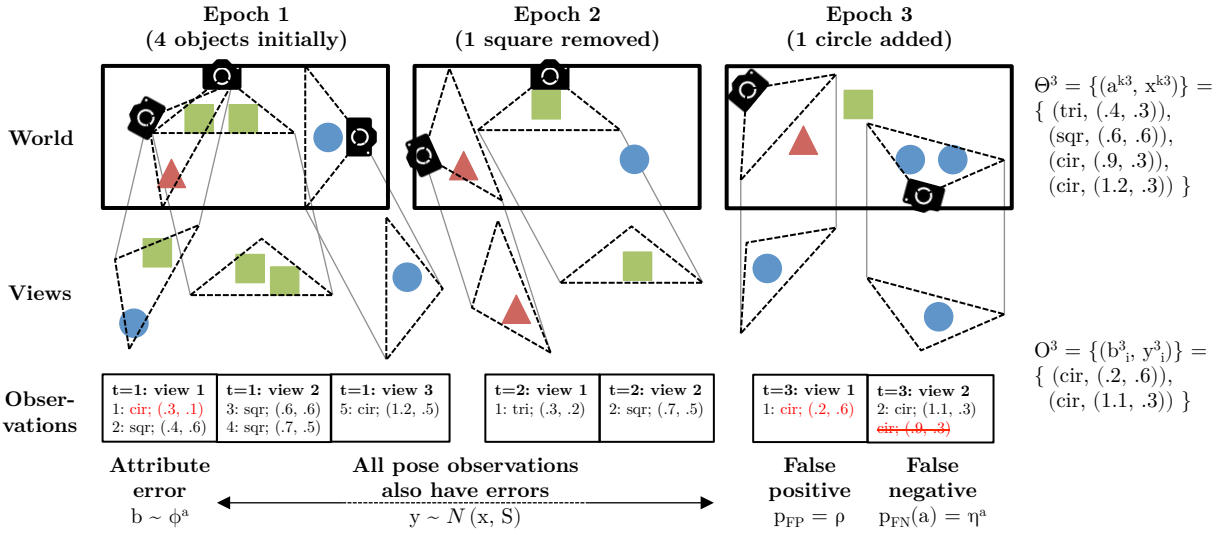
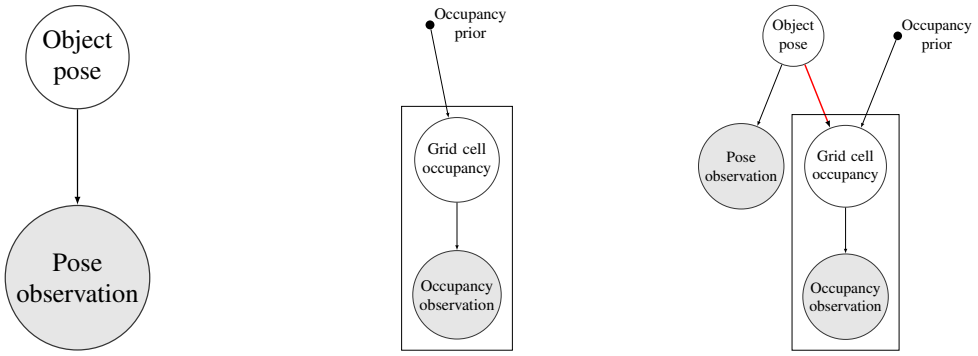


Figure 1-2: An illustration of the world modeling problem. An unknown number of objects exist in the world (top row), and change in pose and number over time (world at each epoch enclosed in box). At each epoch, limited views of the world are captured, as depicted by the triangular viewcones. Within these viewcones, objects and their attributes are detected using black-box perception modules (e.g., off-the-shelf object detectors). In this example, the attributes are shape type (discrete) and 2-D location. The observations are noisy, as depicted by the perturbed versions of viewcones in the middle row. Uncertainty exists both in the attribute values and the existence of objects, as detections may include false positives and negatives (e.g., $t = 3$). The actual attribute detection values obtained from the views are shown in the bottom row ("Observations"); this is the format of input data. Given these noisy measurements as input, the goal is to determine which objects were in existence at each epoch, their attribute values (e.g., Θ^3 in top right), and their progression over time.

(MOT) problems (Luo et al., 2014), which has been well-studied in the computer vision and target-tracking communities. Indeed, conventional approaches such as multiple hypothesis tracking (MHT) (Reid, 1979) and more recent batch methods such as Markov-chain Monte Carlo data association (MCMCDA) (Oh et al., 2009) can be applied directly to our problem. However, we can exploit the (semi-)static nature of our data to reap great computational gains, as we demonstrate against MHT in the static case. Moreover, the semi-static nature of the data actually degrades the performance of MCMCDA. We extended our clustering-based approach for static semantic world modeling to allow clusters to change over time, and developed novel inference algorithms that can efficiently achieve superior performance in semi-static environments.



(a) Object-level **(b)** Metric-level **(c)** Information fusion

Figure 1-3: Graphical model depiction of object and metric spatial information.
(a) Object attributes such as pose are detected and tracked using the methods described in Section 1.2.1.
(b) Occupancy observations are used separately to maintain a standard occupancy grid (Thrun et al., 2005).
(c) Combining these two sources of information is useful (see Figure 1-4); I derived an inference procedure to link the two (red arrow).

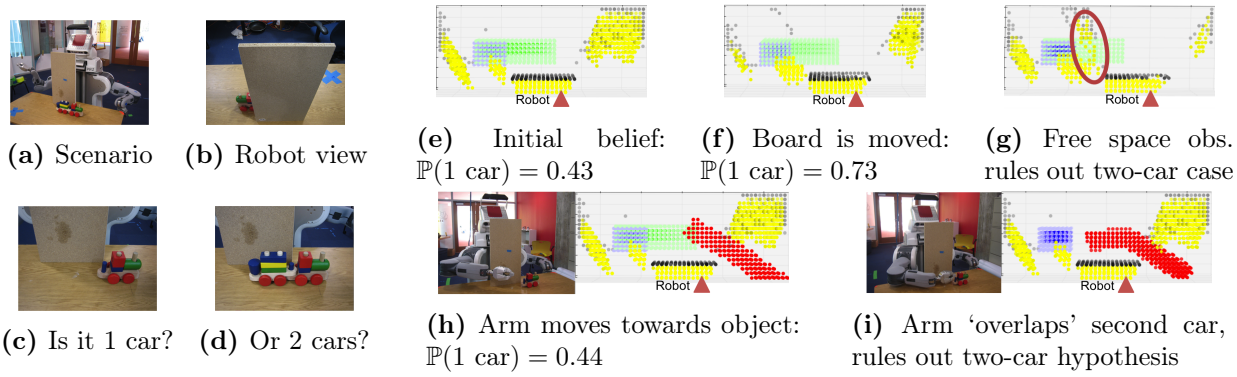
1.2.3 Combining Object and Metric Spatial Information (Wong et al., 2014)

Alas, not all things in the world are objects and attributes. One concept that was lacking in the above work was the notion that objects occupy physical regions of space. The concept of free space, regions that no object overlaps, was also only implicitly represented. It is therefore difficult, in the object-attribute representation, to incorporate absence/'negative' observations, most prominently that observing a region of free space should suggest that no object overlaps that region. On the other hand, this information is handled very naturally in an occupancy grid, but grids cannot incorporate the concept of 'objects' (besides representing them as a collection of cells).

The complementary advantages of these two representations inspired a search for a way to maintain filters of both object and metric information. Because filtering in the joint state involves complex dependencies and is intractable, I instead adopted the strategy of filtering *separately* in the object and metric spaces by using the existing filters. To compensate for the lost dependencies between objects and their geometric realizations, I then developed a way to *merge* the filters on demand as queries about either posterior distribution are made.

Since Moravec and Elfes (1985) pioneered the occupancy grid model of space, occupancy grids have been used extensively in robotics, most notably in mapping. These maps have paved the way for tasks such as navigation and motion planning, in which knowledge of free and occupied spaces is sufficient for success. However, as we move to tasks that require richer interaction with the world, such as locating and manipulating objects, occupancy information alone is insufficient.

In the mapping community, there has been recognition that using metric representations only is insufficient. In particular, the rise of topological mapping, and the combination of the two in hybrid metric-topological mapping (Thrun, 1998) suggests the utility of going beyond metric representations. These hybrid representations have been successfully applied in tasks such as navigation (Konolige et al., 2011). In the related field of semantic mapping (e.g., Kuipers, 2000; Ekvall et al.,



(a) Scenario **(b)** Robot view **(c)** Is it 1 car? **(d)** Or 2 cars?

(e) Initial belief: $\mathbb{P}(1 \text{ car}) = 0.43$ **(f)** Board is moved: $\mathbb{P}(1 \text{ car}) = 0.73$ **(g)** Free space obs. rules out two-car case

(h) Arm moves towards object: $\mathbb{P}(1 \text{ car}) = 0.44$ **(i)** Arm 'overlaps' second car, rules out two-car hypothesis

Figure 1-4: A 3-D demonstration on a PR2 robot. Plots show occupancy grids with 1m × 0.4m × 0.2m volume, containing 10^4 cubes of side length 2cm, with the final (vertical) dimension projected onto the table. Colors depict occupancy type/source: Yellow = free space observation; Black = occupancy observation; Blue = inferred occupancy from one-car train; Green = inferred occupancy from two-car train; Red = occupied by robot in its current state. In this projection, the robot is situated at the bottom center of the plot, facing 'upwards'; the black line observed near the bottom corresponds to the board.
(a)-(b) A toy train is on a table, but only part of the front is visible to the robot.
(c)-(d) This is indicative of two possible scenarios: the train has one car or two cars; there is in fact only one car.
(e)-(g) One way to determine the answer is to move the occluding board away. This reveals free space where the second car would have been (circled in (e)), hence ruling out the two-car case.
(h)-(i) Another way is to use the robot arm. If the arm successfully sweeps through cells without detecting collision, the cells must have originally been free and are now occupied by the arm. Sweeping through where the second car would have been therefore eliminates the possibility of the train being there.
Video: `http://lis.csail.mit.edu/movies/ICRA14_1678_VI_fi.mp4`

2007; Pronobis and Jensfelt, 2012; Liu and von Wichert, 2014), topological information is typically extracted from metric layers (occupancy grids). As described in Section 1.1, some works in semantic mapping do place greater emphasis on the detailed modeling of objects (e.g., Ranganathan and Dellaert, 2007; Wurm et al., 2011; Mason and Marthi, 2012). Wurm et al. (2011) modeled objects as local grid models within a hierarchy of octrees, where the hierarchy was informed by support relations. Mason and Marthi (2012) describe a system that maintains world models by detecting protrusions atop planes and considering them as 'objects'. However, as with the hybrid mapping community, object-based information is rarely propagated back down to the metric level.

There has also been recent interest in fusing object poses with occlusion (Wyffels and Campbell, 2015) and contact information (Koval et al., 2015) respectively.

Combining object-level and metric-level information is useful, as depicted in the example in Figure 1-4. In particular, I identified two ways in which fusion is particularly informative: free space detections strongly indicate that objects cannot be positioned in such regions (Figure 1-4(g)), and object pose detections can be used to infer that overlapping occupancy cells must be occupied. By considering the hypothetical occupancy induced by objects and observing occupancy information that is inconsistent, hypotheses about objects' attributes can be ruled out, and uncertainty reduced. Finally, I have also conducted a brief analysis that justifies our *factor-fuse-forget* filtering framework, showing that it is the more desirable one of two sound filtering possibilities.

Chapter 2

Technical Background

The world is an uncertain place: our perception of it is noisy and incomplete. An agent's degree of uncertainty can affect the value of potential actions; for example, information gathering may be the best option when under great uncertainty (Kaelbling et al., 1998). It is therefore crucial that, when estimating the state of the world, a notion of the *uncertainty* in the estimate is also reported.

Probability theory, based on the axioms of Kolmogorov (1933, 1956), is the calculus of uncertainty adopted in this dissertation. We implicitly estimate a parameter of interest θ by maintaining its *distribution* $\mathbb{P}(\theta)$. Point estimates can be derived from this distribution, given an appropriate cost/loss function (e.g., the most-likely state is the optimal estimate for a 0-1 loss function).

To improve our parameter estimate, we must make a (potentially indirect) observation, resulting in a measurement o. Bayes' rule provides a mechanism for soundly incorporating uncertain evidence (Bayes and Price, 1763)[1]. In modern probability notation, the theorem states:

$$\mathbb{P}(\theta \,|\, o) = \frac{\mathbb{P}(o \,|\, \theta)\ \mathbb{P}(\theta)}{\mathbb{P}(o)} \propto \mathbb{P}(o \,|\, \theta)\ \mathbb{P}(\theta) \tag{2.1}$$

In words, the *posterior* distribution of the parameter θ given observation o (left) is proportional to the product of the observation's *likelihood* and the *prior* distribution on θ (right). A distribution is obtained after the product on the right is normalized over the space of valid parameters. The likelihood is the probability of measuring a specific value of o, assuming the world behaved according to a specific value of θ. The prior is the distribution of θ, either *ab initio*, or, more often, after incorporating some other measurements (but not o).

Bayes' rule is a simple yet profound consequence of probability theory. All of the computation in this dissertation is essentially dedicated to probabilistic inference by 'turning the Bayesian crank'.

I shall no longer dwell on the fundamentals at this point; the interested reader is referred to numerous excellent texts on Bayesian modeling and inference (e.g., Box and Tiao, 1973; Bernardo and Smith, 1994; Robert, 2007; Gelman et al., 2013). The remainder of this chapter is devoted to introducing advanced technical models and tools that will be used in this dissertation.

[1] McGrayne (2011), in her popular historical account on Bayes' rule, argues for the name "Bayes-Price rule" under modern convention, in recognition of Richard Price's role in discovering and disseminating Rev. Thomas Bayes' work.

2.1 Data Association: A Select Review

Most applications of Bayesian inference do not have ambiguity in the *association* between observations and parameters: if an object is in the world, and we get multiple measurements of its location, Bayes' rule informs us about how to update our distribution on the object's location. However, if there are two nearby objects that are identical besides their locations, and we get two noisy location measurements, then it is unclear which observation was generated by detecting which object. Further ambiguity is possible if we get one measurement (due to a missed detection / false negative), if we get three (due to a spurious measurement / false alarm / false positive), and so on.

This is the *data association* problem.

More formally, suppose the object locations are independent and have values $\Theta = \{\theta^1, \theta^2\}$, and the obtained measurements are $O = \{o^a, o^b\}$ (typically the ordering in both objects and observations are arbitrary). If we knew the association of observations to objects, which is a mapping from indices $\{a, b\}$ to $\{1, 2\}$, then we can update the parameters accordingly using Bayes' rule:

$$\begin{aligned} Z = \{a \mapsto 1, b \mapsto 2\} : \mathbb{P}\left(\theta^1 \,|\, O, Z\right) = \mathbb{P}\left(\theta^1 \,|\, o^a\right) \;;\; \mathbb{P}\left(\theta^2 \,|\, O, Z\right) = \mathbb{P}\left(\theta^2 \,\middle|\, o^b\right) \\ \Rightarrow \mathbb{P}\left(\Theta \,|\, O, Z\right) \propto \mathbb{P}\left(\theta^1 \,|\, o^a\right)\; \mathbb{P}\left(\theta^2 \,\middle|\, o^b\right)\; \mathbb{P}\left(\Theta\right) \end{aligned} \tag{2.2}$$

$$\begin{aligned} Z = \{a \mapsto 2, b \mapsto 1\} : \mathbb{P}\left(\theta^1 \,|\, O, Z\right) = \mathbb{P}\left(\theta^1 \,\middle|\, o^b\right) \;;\; \mathbb{P}\left(\theta^2 \,|\, O, Z\right) = \mathbb{P}\left(\theta^2 \,|\, o^a\right) \\ \Rightarrow \mathbb{P}\left(\Theta \,|\, O, Z\right) \propto \mathbb{P}\left(\theta^1 \,\middle|\, o^b\right)\; \mathbb{P}\left(\theta^2 \,|\, o^a\right)\; \mathbb{P}\left(\Theta\right) \end{aligned} \tag{2.3}$$

Notice that the posterior distribution is now conditioned on the association variable Z as well, and the resulting parameter probability value is in general different for different associations. Since the association is unknown, the Bayesian solution is to take the expectation over Z:

$$\mathbb{P}\left(\Theta \,|\, O\right) = \sum_Z \mathbb{P}\left(\Theta, Z \,|\, O\right) = \sum_Z \mathbb{P}\left(\Theta \,|\, O, Z\right)\; \mathbb{P}\left(Z \,|\, O\right) \tag{2.4}$$

$$\propto \left[\mathbb{P}\left(\theta^1 \,|\, o^a\right)\; \mathbb{P}\left(\theta^2 \,\middle|\, o^b\right) + \mathbb{P}\left(\theta^1 \,\middle|\, o^b\right)\; \mathbb{P}\left(\theta^2 \,|\, o^a\right)\right]\; \mathbb{P}\left(\Theta\right) \tag{2.5}$$

The final line follows from the typical assumption that all association mappings are equally likely *a priori*, and the only feasible associations are the two shown above.

Although the summation above appears manageable, it often breaks *conjugacy* properties that are computationally desirable in Bayesian inference and filtering. The number of terms in the summation also grows with the number of mappings Z, which is combinatorial in the number of objects and observations. Furthermore, the mappings may not be perfect matchings as in the above; there may be false positives, false negatives, and out-of-view objects as well, which leads to uncertainty in the true number of objects. The variable dimensionality in Θ also raises serious issues about whether the posterior distribution gives well-defined point estimates (consider: what is the posterior mean of Θ?); this additional wrinkle motivates the philosophy described in Section 2.1.3.

The kernel of the data association problem is in the above summation.

The data association problem was historically motivated by target tracking in surveillance and defense applications (imagine multiple aircraft generating moving blips on a RADAR screen), and is sometimes known synonymously as the *multiple target tracking* (MTT) problem. Bar-Shalom and Fortmann (1988) provide a comprehensive overview of the foundations and early methods. This text was followed by numerous handbooks of methods and edited volumes of applications, recently culminating in the tome by Bar-Shalom et al. (2011). Stone et al. (2014) gives a lighter and slightly different perspective that is more modern and Bayesian in nature.

Due to the diversity of MTT applications and the relative difficulty in computing their solutions, an immense set of algorithms have been proposed. Not all are applicable to the world modeling problem that is the focus of this dissertation. For example, nearest-neighbor methods typically ignore the ambiguity in association by assuming the most likely one; other popular methods such as the joint probabilistic data association filter (JPDAF) assume a fixed number of targets, which is not true in our problem (Bar-Shalom and Fortmann, 1988). Below, I describe three types of methods: multiple hypothesis tracking (MHT), Markov-chain Monte Carlo (MCMC), and random finite sets (RFS). The first two will provide motivation for our models, and will act as comparisons against our algorithms in subsequent chapters. RFS-based methods, of which the probability hypothesis density (PHD) filter is perhaps the most well-known exemplar, is only discussed here for context.

2.1.1 Multiple hypothesis tracking (MHT)

A principled approach to MTT when the number of tracks is unknown is multiple hypothesis tracking (MHT) (Reid, 1979). MHT considers the tree of all possible association hypotheses, branching on the possible tracks that each measurement can correspond to. At each node, the possibilities for each measurement include assigning to an existing track, initiating a new track, or declaring a false positive. The tree is effectively enumerating the summands of Equation 2.4, in some pre-determined order (often chronological). Each node contains an unnormalized probability value that acts as the prior for the next layer. When a new layer is grown, a child's value is found by multiplying its observation's likelihood with its parent's value, thereby performing one step in the Bayes' rule computation. Normalizing the values of the leaves gives the distribution of interest, $\mathbb{P}(\Theta \mid O)$.

Using the previous example involving two existing objects and two observations, where we process o^a and then o^b, and where we now allow false positives (FP) and new objects (New, which becomes object index 3), we can enumerate the tree of valid associations:

$$o^a : \{a \mapsto 1\}, \{a \mapsto 2\}, \{a \mapsto \text{New}\}, \{a \mapsto \text{FP}\} \tag{2.6}$$

$$\begin{aligned} o^b : &\{a \mapsto 1, b \mapsto 2\}, \{a \mapsto 1, b \mapsto \text{New}\}, \{a \mapsto 1, b \mapsto \text{FP}\} \\ &: \{a \mapsto 2, b \mapsto 1\}, \{a \mapsto 2, b \mapsto \text{New}\}, \{a \mapsto 2, b \mapsto \text{FP}\} \\ &: \{a \mapsto 3, b \mapsto 1\}, \{a \mapsto 3, b \mapsto 2\}, \{a \mapsto 3, b \mapsto \text{New}\}, \{a \mapsto 3, b \mapsto \text{FP}\} \\ &: \{a \mapsto \text{FP}, b \mapsto 1\}, \{a \mapsto \text{FP}, b \mapsto 2\}, \{a \mapsto \text{FP}, b \mapsto \text{New}\}, \{a \mapsto \text{FP}, b \mapsto \text{FP}\} \end{aligned} \tag{2.7}$$

Each row in the o^b list above corresponds to one possible assignment for o^a; false negatives (FN) are implicit (when no observation is mapped to an existing object). Also, we have discarded associations where both observations are mapped to the same object (in the same 'scan' only); this is a common assumption in data association problems (Bar-Shalom and Fortmann, 1988), which I refer to as the 'cannot-link constraint' (CLC) (unfortunately, there is no common name). For each mapping in the list, a likelihood value / posterior parameter distribution can be computed.

In general, because each path in the tree leads to a different posterior distribution, we must keep around the entire tree to maintain a sufficient statistic for the multiple hypothesis filter; compression (by combining paths) is lossy. However, due to the number of targets and measurements involved, maintaining the entire tree (and hence the exact posterior distribution) is exponentially expensive and intractable for any non-trivial branching factor. As a result, practical implementations of MHTs must use one of many proposed heuristics (e.g., Kurien (1990); Cox and Hingorani (1996)), typically pruning away all but the few most-likely branches in the association tree.

A popular pruning technique known as "N-scan back" maintains the full tree for only the most recent N 'scans' (times/views), and only keeps the most likely path going further back in time. This can be viewed as delayed filtering by a window of N scans, where data association commitments are only made final after N steps, when hopefully further disambiguating information has been acquired. Nevertheless, to maintain tractability, N must be relatively small, and correct associations that happen to appear unlikely at the moment may be lost. This is especially true in cases (such as ours) where targets are not in constant motion, and may remain ambiguous over many scans.

2.1.2 Markov-chain Monte Carlo (MCMC) methods

The MHT strategy is to enumerate all possible associations, then to prune away everything besides the most-likely hypotheses. Instead of attempting to evaluate every point in a large space of associations, most of which contains little probability mass, efficient sampling techniques have been proposed that try to only explore high-probability regions. Another way of interpreting this is that instead of evaluating the sum in Equation 2.4, which has an intractable number of summands, we treat it as an expectation of some function $\mathbb{P}(\Theta \mid O, Z)$ over the distribution of associations $\mathbb{P}(Z)$.

Sampling-based schemes also differ in that they are *batch* methods and require storing observations over all scans of interest. Although a storage overhead is paid, batch processing frees up the need to make quick association commitments, which leads to greater reliability and potential computational savings. Although offline in nature, sampling-based methods for data association can potentially be used to complement online methods such as the MHT.

The space of associations is a large and complicated discrete landscape to explore, requiring sophisticated sampling techniques. The Monte Carlo method, in particular Markov-chain Monte Carlo (MCMC), is a successful and widely-applied class of techniques for sampling from difficult spaces (Metropolis and Ulam, 1949). Robert and Casella (2004) provide a broad overview of the subject. MCMC methods, such as the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) and Gibbs sampling (Geman and Geman, 1984), generate a sequence of samples

Input: Observations O, Number of desired samples N
Output: Samples of measurement-to-track associations $\left\{Z^{(n)}\right\}_{n=1}^{N}$

1: Initialize all associations to false positives (FP) in $Z^{(0)}$
2: **for** $n := 1$ **to** N **do**
3: $\quad Z' := \text{Proposal}(Z^{(n-1)})$ (e.g., Pasula et al., 1999; Dellaert et al., 2003; Oh et al., 2009)
4: $\quad A\left(Z^{(n-1)} \to Z'\right) := \min\left(1,\ \frac{\mathbb{P}(O \mid Z')}{\mathbb{P}\left(O \mid Z^{(n-1)}\right)} \frac{\mathbb{P}\left(Z' \to Z^{(n-1)}\right)}{\mathbb{P}\left(Z^{(n-1)} \to Z'\right)}\right)$
5: $\quad$ Sample $u \sim \text{Unif}(0, 1)$
6: $\quad$ **if** $u < A\left(Z^{(n-1)} \to Z'\right)$ **then**
7: $\quad\quad Z^n = Z'$ (accept)
8: $\quad$ **else**
9: $\quad\quad Z^n = Z^{(n-1)}$ (reject)

Figure 2-1: Metropolis-Hastings (MH) algorithm for performing data association.

from a Markov chain (often a random walk), whose stationary distribution is constructed to be the distribution that one wishes to draw samples from. In Bayesian inference, this target distribution is typically the posterior distribution. New candidate samples are drawn from a proposal distribution that is typically parameterized by the last sample in the sequence, and are accepted with some probability that ensures the correctness of the construction (as long as the constructed Markov chain is ergodic). Because proposals are typically local in nature, consecutive samples are dependent, and it may take many samples to fully explore the space. Nevertheless, MCMC is often the only effective sampling technique available in such problems.

MCMC methods for sampling association matchings and tracks have been explored by Pasula et al. (1999) for traffic surveillance and by Dellaert et al. (2003) for structure-from-motion. In the former, samples were proposed by swapping assignment pairs, such as $\{a \mapsto 1, b \mapsto 2\} \to \{a \mapsto 2, b \mapsto 1\}$. Although simple in nature, such moves are sufficient to ensure a correct MCMC construction, assuming no tracks are introduced or terminated. However, due to the locality of proposals, Dellaert et al. (2003) found that single swaps alone can lead to slow convergence rates, and therefore developed "chain flipping", which proposes sequences of reassignments reminiscent to those in weighted assignment algorithms. The form of these MCMC sampling algorithms is shown in Figure 2-1; the major difference is in the proposal distribution (step 3), and the subsequent calculation of transition probabilities (step 4, forward $\mathbb{P}(Z \to Z')$ and reverse $\mathbb{P}(Z' \to Z)$).

More recently, Oh et al. (2009) generalized the above work by considering an even wider class of transition moves during sampling, and provided theoretical bounds on the mixing (convergence) time of the resulting sampling algorithm, MCMCDA. The algorithm first constructs a neighborhood tree that connects each observation with a set of plausible successors (similar to gating in MHT, but potentially over multiple scans). Association tracks are formed from path segments within this tree. Proposals are generated by the following five forward/reverse move pairs:

- Birth: An unassigned path segment forms a track / Death: An existing track is unassigned
- Split: A track is split randomly / Merge: Two tracks are connected, if new edge is valid

- Extend: Observations are appended to a track / Reduce: A track's end is cut off
- Update: Individual observations are added/removed from a track
- Switch: A pair of edges in the middle of two tracks is swapped (same as Pasula et al. (1999))

A nice illustration of these moves is provided in Figure 3 in Oh et al. (2009).

Because only a small space of likely associations is frequently sampled, and all measurement associations are repeatedly considered (unlike MHT with pruning), MCMCDA empirically outperforms MHT both in efficiency and accuracy, especially in environments with heavy detection noise. We expect this to generally be the case for MCMC-based methods for data association, as long as the proposal distribution is sufficiently powerful to efficiently explore the space of associations.

2.1.3 Random finite sets (RFS)

A fundamentally different approach to multitarget tracking problems is motivated by two unsettling mathematical observations:

- The states of targets Θ and their measurements O are typically formulated as random vectors with an arbitrary ordering that does not exist in reality.
- There is often uncertainty in the number of targets, i.e., the dimensionality of Θ. Because the posterior distribution $\mathbb{P}(\Theta \mid O)$ is a variable-dimension entity, point estimates such as the posterior mean and the maximum *a posteriori* (MAP) are not well-defined.

Goodman et al. (1997) propose that a more suitable mathematical object is a *random finite set* (RFS), whose cardinality is also random, and is by definition unordered.

Mahler (2007, 2014) expands on this agenda in great detail, transforming results from a rigorous foundation for RFSs based in abstract point-process theory (Daley and Vere-Jones, 1988) into a concrete paradigm of finite set statistics (FISST), which in turn leads to new multisource-multitarget tracking algorithms. A more approachable philsophical summary and mathematical tutorial of this line of work can be found in Mahler (2004) and Mahler (2013) respectively. The basic philosophy is to draw parallels with the much better-understood case of single-source, single-target problems. The state is treated as a single RFS, with transitions governed by "multitarget motion models" and observations made according to "multitarget measurement models". The multitarget models essentially 'bundle' together multiple single-target models, being careful about the variable dimensionality (due to FPs and FNs), and has an abstract input-output interface solely in the language of RFSs. "Being careful" means that, under the hood, all association possibilities for single states are still considered; however, that is abstracted away by a new "multitarget calculus".

The basic probabilistic object in FISST is the *belief mass function*, a multitarget analog of the conventional probability mass function:

$$\beta(X) \triangleq \mathbb{P}(Y \subseteq X) \tag{2.8}$$

A multitarget analog of the conventional probability density function can be defined via the notion of a set integral (over region S):

$$\int_S f(X)\ \delta X \triangleq \sum_{n=1}^{\infty} \frac{1}{n!} \int_{S\times\cdots\times S} f(\{x_1,\ldots,x_n\})\ \mathrm{d}x_1\cdots\mathrm{d}x_n \tag{2.9}$$

The multitarget density function of an RFS is the function whose set integral is the belief mass function of the RFS. This operation is well-defined and serves as the multitarget/set analog of the derivative. Note that the $\frac{1}{n!}$ term prevents overcounting of RFS instantiations in $S\times\cdots\times S$. Also, the density function $f(X)$ has argument-dependent units $u^{-|X|}$, for a generic unit u.

To see how a multitarget Bayes filter operates, it is instructive to look at the form of the multitarget motion/observation models. Consider taking a measurement on a single target with state x_i with some probability of detection $0 < p_{\mathrm{D}}(x_i) < 1$. Then the possible values of this random *set* event Υ_i are $\emptyset$, which occurs with probability $p_{\mathrm{D}}(x_i)$, and $\{y_i\}$ (a single measurement), which occurs with (conventional) density $p_{\mathrm{D}}(x_i)\ f(y_i \mid x_i)$. The belief mass function of Υ_i is:

$$\beta_{\Upsilon_i}(Y) = [1 - p_{\mathrm{D}}(x_i)] + p_{\mathrm{D}}(x_i) \int_Y f(y_i \mid x_i)\ \mathrm{d}y_i \tag{2.10}$$

Since the conditional density above is the conventional one, the integral is conventional too.

The event Υ_i pertains to a single target; we now need to bundle multiple instances for additional targets. For a given state set $X = \{x_1,\ldots,x_n\}$, the multitarget observation RFS event is:

$$\Xi = \Upsilon_1 \cup \cdots \cup \Upsilon_n \cup C \tag{2.11}$$

C is the set of clutter/FP measurements, and is usually modeled as a Poisson RFS. Assuming that measurements are generated independently, the belief mass function of Ξ is:

$$\beta_{\Xi}(Y \mid X) = \beta_{\Upsilon_1(x_1)}(Y)\ \cdots\ \beta_{\Upsilon_n(x_n)}(Y)\ \beta_C(Y) \tag{2.12}$$

Taking the set derivative of β_{Ξ} gives the multitarget observation density (Mahler (2007)):

$$f(Y \mid X) = \left[e^{-\lambda} \prod_{y\in Y} \lambda c(y)\right] \left[\prod_{x\in X} (1 - p_{\mathrm{D}}(x))\right] \left[\sum_{Z} \prod_{(i,j):Z(i)=j} \frac{p_{\mathrm{D}}(x_i)\ f(y_i \mid x_j)}{(1 - p_{\mathrm{D}}(x_i))\ \lambda c(y_i)}\right] \tag{2.13}$$

Here λ is the rate parameter of the Poisson RFS, and $c(y)$ is the likelihood of FP measurements. The final term is a summation over all valid associations Z; for each association, the product is taken over all observation-object index pairs where an observation y_i is mapped to a hypothesized target x_j. This summation term is similar in spirit to the data association summation in Equation 2.4. Although users of FISST methods often claim that the methods avoid data association, this is only true in the sense that no explicit associations are represented (unlike MHT/MCMC methods); however, the difficult problem of marginalizing out the latent association variable Z remains.

The multitarget motion model $f(X' \,|\, X)$ has a similar form, except that the probability of detection p_{D} is replaced by the probability of target survival p_{S}, the clutter Poisson RFS β_C is replaced by target birth Poisson RFS β_B, and the conventional observation density $f(y \,|\, x)$ is replaced by a transition density $f(x' \,|\, x)$. To complete the multitarget recursive Bayes filter:

$$f\left(X' \,\middle|\, Y, Y'\right) = \frac{f(Y' \,|\, X') \left[\int f(X' \,|\, X)\ f(X \,|\, Y)\ \delta X\right]}{\int f(Y' \,|\, X') \left[\int f(X' \,|\, X)\ f(X \,|\, Y)\ \delta X\right] \delta X'} \tag{2.14}$$

This is completely analogous to the single-target case, with the interpretation now that the variables are all RFSs, f is a multitarget probability density function, and "$\int$" is a set integral.

Although the notation was deliberately chosen by Mahler to draw connections to the single-target case, the form of multitarget likelihood functions (Equation 2.13) shows that the computational details are much more complicated. In the single-target case, the celebrated Kalman filter provides exact, closed-form, and optimal filtering of a state variable that is subject to linear transition and observation models perturbed with Gaussian noise; the estimated state is normally-distributed Kalman (1960). There is no such analog in the multitarget case. In fact, there is no known non-trivial multitarget density that admits closed-form recursive filtering.

The closest one can get to nice analytical filtering is via the probability hypothesis density (PHD) filter, and is perhaps the most popular RFS-based filter so far Mahler (2003). The PHD $h(x)$ is the first-moment approximation to the multitarget density function, characterized by:

$$\int_S h(x \,|\, Y)\ \mathrm{d}x = \int_S |X \cap S|\ f(X \,|\, Y)\ \delta X = \text{Expected number of targets in } S \tag{2.15}$$

Note that the first integral is of the conventional kind, whereas the second is a set integral. h is also referred to as the *intensity* function, and is a measure of the spatial density of targets; as the characterization above shows, the PHD does *not* integrate to 1, and is therefore *not* a probability density. Instead, it integrates to the expected number of targets in the domain.

The 'classical' PHD filter makes additional assumptions (not necessarily true of all PHD filters):

- Independence of target motions, and conditonal independence of measurements given states.
- The clutter (β_C) and birth (β_B) processes are Poisson RFSs.
- Target detection (p_{D}) and target survival (p_{S}) are Bernoulli events.

These assumptions are often made even in FISST methods beyond the PHD, and were already made earlier when illustrating the multitarget observation density (Equation 2.13). Under these assumptions, the recursive PHD filter simplifies to:

$$h^+\left(x' \,\middle|\, Y\right) = b(x') + \int p_{\mathrm{S}}(x)\ f\left(x' \,\middle|\, x\right)\ h(x \,|\, Y)\ \mathrm{d}x \tag{2.16}$$

$$h'\left(x' \,\middle|\, Y, Y'\right) = \left[1 - p_{\mathrm{D}}\left(x'\right)\right] h^+\left(x' \,\middle|\, Y\right) + \sum_{y' \in Y'} \frac{p_{\mathrm{D}}(x')\ f(y' \,|\, x')\ h^+(x' \,|\, Y)}{\lambda c(y') + \int p_{\mathrm{D}}(x)\ f(y' \,|\, x)\ h^+(x \,|\, Y)\ \mathrm{d}x} \tag{2.17}$$

The first equation corresponds to the prediction step. Given a previous PHD $h(x \mid Y)$, we consider transitions from x to x', and decrease the intensity by a factor of p_{S}, depending on how likely a target at x may survive. An independent birth PHD b is added to the propagated PHD, resulting in the predictive PHD $h^{+}(x' \mid Y)$. In the second equation, we incorporate new observations Y' using the multitarget Bayes' rule. The first term accounts for the case where a target may have been missed (hence the factor of $1 - p_{\mathrm{D}}$). The second term is basically Bayes' rule: for each observation y', its likelihood of being generated by x' is added to the PHD, after being normalized against all other possibilities (including by clutter, c). The normalization ensures that the characterizing property of the PHD, that its integral is equal to the expected number of targets, is maintained.

Even with the above simplifications, the PHD filter *still* does not have closed-form solutions in general! The PHD characterization admits a large degree of flexibility, and we do not expect the class of PHDs to be easily parameterizable. PHDs can be multi-modal (if targets appear in separated regions) with arbitrarily many modes, which is difficult to capture in typical classes of function. In general, the PHD function itself needs to be approximated, typically using sequential Monte Carlo methods (Vo et al., 2005) or with a mixture of Gaussians (Vo and Ma, 2006).

Within the context of localization and mapping, both types of approximations have been used for PHD filters. Mullane et al. (2011) used a Gaussian-mixture (GM-) PHD to perform feature-based SLAM; a succinct overview of the approach is given by Adams et al. (2014) (in particular, see Figures 3 and 4). The issue with GM-PHDs is that, given a new scan of observations Y, the number of Gaussian components grows by a factor of $|Y|$; in general, each observation evolves the existing components in a different way and causes splitting. Since the number of components grows exponentially with the number of scans, a pruning strategy that removes/combines smaller components is necessary. Nevertheless, their PHD-based SLAM outperforms the conventional MHT-FastSLAM approach. Similarly, Kalyan et al. (2010) used a particle-filter PHD approximation and showed improvement against FastSLAM.

PHD and FISST methods have also been useful for various localization tasks. Dames and Kumar (2015) used a particle representation of the PHD to localize an unknown number of fixed targets using bearing-only measurements. Dames et al. (2014) have also attempted to use the full FISST multitarget Bayes filter on the same task, using a quadtree-based representation that allows adaptive discretization of the environment. Their use of FISST allows the filter to capture higher-order moments, including correlations in target locations. In contrast, Atanasov et al. (2016) considered localizing a robot (instead of external targets) against a known map of landmarks. A particle-based representation of the multitarget density is used. To handle the implicit data assocation summation in the multitarget likelihood (Equation 2.13), the authors point out an elegant connection between the summation over matchings and computing matrix permanents, the latter of which is well-studied and has numerous exact and approximate algorithms (but is a #P-complete problem in general).

Random finite set methods are built on a rigorous mathematical foundation, and have been successfully applied in an increasing number of problems. I am cautiously optimistic that RFSs will provide a unifying framework for multitarget tracking, data association, and sensor fusion.

2.2 Bayesian Nonparametric Clustering

We now switch to clustering, a subject that seems completely different on the surface, and is situated in the fields of statistics and machine learning. In subsequent chapters, I will draw connections between clustering and data association. At a high level, inferring underlying clusters within data often requires solving a data association problem (although it is usually not referred to in this way): figuring out which cluster generated the each data point. One can also consider static targets in the multitarget tracking problem as clusters, and target tracks as dynamic / evolving clusters. This view will be presented in much greater detail in Chapters 3 and 4.

The idea behind cluster analysis is that, when presented with a large amount of data, we assume that it was generated by a relatively small number of phenomena, and we would like to extract this underlying low-dimensional structure from the data. For example, documents typically only contain content (words) pertaining to a few topics (Blei et al., 2003). The range of approaches devoted to clustering is a testament to the problem's importance; see Jain (2010) for an overview. We focus on the class of distribution-based clustering methods, in particular using *mixture models*.

2.2.1 Finite mixture models

Mixture modeling assumes that the world consists of multiple components, each with its own state θ_j, and each component is capable of generating observations $O = \{o_i\}$ within the same space. Observations from mixture component j have distribution $f(o\,;\,\theta_j)$, where f is some probability density function specified by the model. Component densities may differ in both parameters *and* class (e.g., some are Gaussian, others are uniform). Given observations O, the goal is to infer the mixture component parameters $\Theta = \{\theta_j\}$, and possibly also their number K.

The challenge is that information about which component generated each observation is missing: there is a data association problem.

For now, assume we have decided on a mixture model with K components; we will delve into how to choose this number in the next section. Measurements have the following distribution:

$$\mathbb{P}(o \,|\, \Theta, \pi) = \sum_{j=1}^{K} \pi_j \; f(o\,;\,\theta_j) \tag{2.18}$$

The vector π are the mixture weights, and is a discrete distribution.

In Bayesian mixture modeling, we specify prior distributions on Θ and π, and perform Bayesian inference. A typical generative model for N mixture model observations is:

$$\begin{aligned}
\pi &\sim \mathrm{Dir}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right) \\
\theta_j &\sim H(\theta) \;\;,\;\; j \in [K] \\
z_i &\sim \mathrm{Categorical}(\pi) \;\;,\;\; i \in [N] \\
o_i &\sim f(o\,;\,\theta_{z_i}) \;\;,\;\; i \in [N]
\end{aligned} \tag{2.19}$$

The prior on π is a K-parameter Dirichlet distribution, which is a distribution over the $(K-1)$-dimensional simplex (i.e., it is a distribution over K-component discrete distributions). The prior on θ is some arbitrary density H, which is often chosen to be conjugate to the component density f to allow efficient inference. $Z = \{z_i\}$ are the latent cluster assignment / association variables, one for each observation, which specifies the component index j to which measurement i belongs. Each assignment is drawn from the K-component distribution π (which is known formally as a categorical or 'multinoulli' distribution).

If the parameters Θ and mixture weights π were known, we can find the posterior distribution for each assignment variable z_i using Bayes' rule:

$$\mathbb{P}\left(z_i = j \,|\, o_i\,;\, \Theta, \pi\right) = \frac{\mathbb{P}\left(o_i \,|\, z_i = j\,;\, \Theta, \pi\right)\ \mathbb{P}\left(z_i = j\,;\, \Theta, \pi\right)}{\sum_{k=1}^{K} \mathbb{P}\left(o_i \,|\, z_i = k\,;\, \Theta, \pi\right)\ \mathbb{P}\left(z_i = k\,;\, \Theta, \pi\right)} = \frac{f\left(o_i\,;\, \theta_j\right)\ \pi_j}{\sum_{k=1}^{K} f\left(o_i\,;\, \theta_k\right)\ \pi_k} \tag{2.20}$$

Since the parameters are unknown and we actually want to infer them, the above probability must be marginalized over the posterior distributions of Θ and π.

Suppose we already have made previous observations O and have found their cluster assignments Z. Then it can be shown that the posterior distributions on the parameters are:

$$\pi \mid O, Z \sim \mathrm{Dir}\left(\frac{\alpha}{K} + N_1, \ldots, \frac{\alpha}{K} + N_K\right) \tag{2.21}$$

$$\theta_j \mid O, Z \sim f\left(\theta \,|\, O|_{i:z_i=j}\right) \propto \left[\prod_{i:z_i=j} f\left(o_i \,|\, \theta\right)\right] H\left(\theta\right) \tag{2.22}$$

$N_j \triangleq |\left\{i : z_i = j\right\}|$ is the number of observations assigned to cluster j; clearly, $N = \sum_{j=1}^{K} N_j$.

Given a new observation o', its assignment variable z' has the following distribution:

$$\mathbb{P}\left(z' = j \,\middle|\, o', O, Z\right) \propto \mathbb{P}\left(o' \,\middle|\, z' = j, O, Z\right)\ \mathbb{P}\left(z' = j \,\middle|\, O, Z\right) \tag{2.23}$$

The two terms on the right resemble the terms in the numerator of Equation 2.20, and can be found by marginalizing out the posterior parameter distributions:

$$\begin{aligned}
\mathbb{P}\left(o' \,\middle|\, z' = j, O, Z\right) &= \int \mathbb{P}\left(o', \theta_j \,\middle|\, z' = j, O, Z\right)\ \mathrm{d}\theta_j = \int \mathbb{P}\left(o' \,\middle|\, \theta_j\right)\ \mathbb{P}\left(\theta_j \,|\, O, Z\right)\ \mathrm{d}\theta_j \\
&= \int f\left(o'\,;\, \theta_j\right)\ f\left(\theta_j \,|\, O|_{i:z_i=j}\right)\ \mathrm{d}\theta_j \qquad (2.24) \\
\mathbb{P}\left(z' = j \,\middle|\, O, Z\right) &= \int \mathbb{P}\left(z' = j, \pi \,\middle|\, O, Z\right)\ \mathrm{d}\pi = \int \mathbb{P}\left(z' = j \,\middle|\, \pi\right)\ \mathbb{P}\left(\pi \,|\, O, Z\right)\ \mathrm{d}\pi \\
&= \int \pi_j\ \mathrm{Dir}\left(\frac{\alpha}{K} + N_1, \ldots, \frac{\alpha}{K} + N_K\right)\ \mathrm{d}\pi = \frac{\frac{\alpha}{K} + N_j}{\alpha + N} \qquad (2.25)
\end{aligned}$$

The final expression is the mean of the j'th component of a Dirichlet distribution. The first equation is the posterior predictive distribution for observations generated by the j'th mixture component.

In practice, we do not even know the exact cluster assignments Z of previous observations. To

perform exact inference, we must marginalize over all possible latent assignments Z, but that leads to a difficult data association summation, as discussed in the previous section of this chapter. We often need to resort to approximate inference methods. For example, expectation-maximization (EM) iteratively performs 'soft' cluster assignments, and assuming the maximum-likelihood parameters under these temporary partial assignments (Dempster et al., 1977). Partial assignments are made by allowing observations to contribute to multiple mixture components, weighted by Equation 2.20 (using the parameters found at the most recent iteration). This procedure provably improves the observed data likelihood, and converges to a local optimum. However, EM only finds a point estimate of the parameters. Another approach to approximate inference is by *sampling*, typically in the space of cluster assignments Z, as covered in Section . In particular, in this dissertation I will mostly use Gibbs sampling (Geman and Geman, 1984), which repeatedly samples assignments from their conditional distributions (given all other assignments), as in Equation 2.23.

Texts such as Bishop (2006) and Murphy (2012) provide further background on mixture models, as well as in-depth coverage of Bayesian inference algorithms.

2.2.2 Dirichlet process mixture models (DPMM)

So far, we have assumed a fixed number of mixture components K. However, this number is typically not immediately apparent given the data, and is user-specified. An inappropriate choice of K can lead to poor clustering results. Additionally, in the context of online clustering, the number of relevant phenomena in the world does not always stay fixed; the appropriate number of clusters may change over time. Thus pre-specifying and fixing the value of K may not be a robust solution.

The Bayesian philosophy suggests that for every quantity that we are uncertain about, we should consider its distribution (prior) and perform inference on it. In the case of mixture models, that means letting K be a variable, and inferring its posterior distribution from data. This simple move raises an interesting mathematical issue: how do we handle the dimensionality of the mixture weights π and the number of mixture component parameters θ, both of which depend on K?

One answer is to allow all of them be (countably) infinite, so that each observation, both past and future, could potentially be assigned to a new cluster component. This idea of infinity at first sounds daunting, and seems to violate the spirit of clustering (we want to group observations together). In practice, any clustering problem will only involve a finite amount of data (although it may grow in size), and we can avoid ever having to instantiate infinite mathematical entities. The degree to which different observations are clustered together is also adjustable in models. The idea is to let models 'discover' the appropriate number of clusters from the data.

Such models are known as Bayesian nonparametric models (Ghosh and Ramamoorthi, 2003; Hjort et al., 2010). They are nonparametric in the sense that the number of *instantiated* parameters is not fixed, where instantiated means that the parameter is involved in the description of some existing observation. However, they are not 'truly' nonparametric in the sense that the model does not have parameters (Wasserman, 2006); in fact, Bayesian nonparametric models have an *infinite* number of parameters, of which a finite number have been instantiated in any practical application.

The Dirichlet process (DP) is a fundamental distribution used in Bayesian nonparametric statistics (Ferguson, 1973; Blackwell and MacQueen, 1973). It is a natural generalization of the Dirichlet distribution to infinite spaces, in that it is a distribution over probability measures G, which are characterized by having Dirichlet-distributed finite marginals for any partition $(A_1, \ldots, A_R)$ of Θ:

$$(G(A_1), \ldots, G(A_R)) \sim \text{Dir}(\alpha H(A_1), \ldots, \alpha H(A_R)) \tag{2.26}$$

Θ is the sample space for G and H, and $G(A_i)$ is the measure on that particular subset of Θ. If G satisfies the above condition, we say that G is DP-distributed: $G \sim \text{DP}(\alpha, H)$, where H is the base measure and α is the concentration hyperparameter. An intuitive interpretation of the condition is that H acts as a 'template' for the DP, and the degree to which resemblance is enforced is governed by α. G is a discrete distribution on Θ with probability one, possibly with infinite support.

DPs serve as a natural prior for parameters in infinite mixture models (Antoniak, 1974; Escobar and West, 1995; Neal, 2000). Indeed, the notation above was chosen to deliberately overlap with the finite case. From a generative perspective, instead of taking K samples from H to use as the mixture component parameters, we take N samples from G, which itself is a sample from $\text{DP}(\alpha, H)$. That is, in theory, we sample as many mixture parameters $\tilde{\theta}_j$ as there are observations. In practice, since G is discrete, its atoms are typically sampled many times, i.e., the corresponding observations belong to the same cluster (since they were generated by the same component parameters).

The generative process for observations from a DP mixture model (DPMM) is:

$$\begin{aligned} G &\sim \text{DP}(\alpha, H) \\ \tilde{\theta}_i &\sim G(\tilde{\theta}) \ , \ i \in [N] \\ o_i &\sim f\left(o\,;\, \tilde{\theta}_i\right) \ , \ i \in [N] \end{aligned} \tag{2.27}$$

However, this description is somewhat opaque and non-constructive; unlike the finite case, where mixture weights π were sampled from a Dirichlet distribution, it is unclear how to sample an infinite discrete distribution G from a DP.

To illustrate the analogy of the DPMM to finite mixture models, the DP can be specified in an equivalent form based on the stick-breaking construction of Sethuraman (1994). The idea is that the sizes of clusters are determined by a random process that first selects some proportion β_1 of the unit interval ('breaks the stick'), where $\beta_1 \sim \text{Beta}(1, \alpha)$, and defines that to be the size of the first cluster (as a proportion π_1). This process is then recursively applied *ad infinitum* to the remaining length-$(1 - \beta_1)$ stick, resulting in a countably infinite subdivision of the unit interval:

$$\begin{aligned} \beta_k &\sim \text{Beta}(1, \alpha) \\ \pi_k &\triangleq \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) = \beta_k \left(1 - \sum_{l=1}^{k=1} \pi_l\right) \end{aligned} \tag{2.28}$$

For each π_k, sample a corresponding θ_k from $H(\theta)$, and let $G \triangleq \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$, where δ_x is the Dirac

delta function. Then it turns out that $G \sim \mathrm{DP}(\alpha, H)$.

The stick-breaking construction shows explicitly how to obtain G. We also see what role α plays: smaller values of α tend to result in larger sticks earlier in the process, hence fewer clusters are preferred. Also, the form of G now resembles a mixture distribution. The weights π_k that result process is usually denoted as a sample from a Griffiths-Engen-McCloskey process: $\pi \sim \mathrm{GEM}(\alpha)$. We can now rewrite the generative process using the construction:

$$\begin{aligned} \pi &\sim \mathrm{GEM}(\alpha) \\ \theta_j &\sim H(\theta) \ , \ j \in [\infty] \\ z_i &\sim \mathrm{Categorical}(\pi) \ , \ i \in [N] \\ o_i &\sim f(o\,;\, \theta_{z_i}) \ , \ i \in [N] \end{aligned} \tag{2.29}$$

The process now resembles the finite case shown in Equation 2.19, although it is still not computationally feasible since it involves generating an infinite number of sticks and parameters.

Much like the Dirichlet distribution, the DP has nice conjugacy properties that supports the computation of certain conditional distributions. In the previous generative process (Equation 2.27), suppose we have observed N instances of $\tilde{\theta}_i$, each generated from some G, which in turn is generated from $\mathrm{DP}(\alpha, H)$. Then the posterior distribution of G and the predictive distribution of $\tilde{\theta}$ is:

$$G \mid \tilde{\theta}_1, \ldots, \tilde{\theta}_N \sim \mathrm{DP}\left(\alpha + N,\ \frac{1}{\alpha + N}\left[\alpha H + \sum_{i=1}^{N} \delta_{\tilde{\theta}_i}\right]\right) \tag{2.30}$$

$$\tilde{\theta}' \mid \tilde{\theta}_1, \ldots, \tilde{\theta}_N \sim \frac{1}{\alpha + N}\left[\alpha H + \sum_{i=1}^{N} \delta_{\tilde{\theta}_i}\right] \tag{2.31}$$

The new base distribution contains $\frac{1}{\alpha+N}$ measure on each $\tilde{\theta}_i$. This shows that if we continue to sample new $\tilde{\theta}$ from G, which we see in the constructive process above is related to sampling from the new base measure, there is non-zero probability that $\tilde{\theta} = \tilde{\theta}_i$ for some previous $i \in [N]$. In fact, as the number of samples N grows, there is increasingly less chance $\left(\frac{\alpha}{\alpha+N}\right)$ that a previously-unseen $\tilde{\theta}$ is sampled. Hence the DP prior is a plausible model for clustering.

Since some of the parameters in $\left(\tilde{\theta}_1, \ldots, \tilde{\theta}_N\right)$ may be duplicates, we collect them into a set $\{\theta_1, \ldots, \theta_K\}$ (without duplicates), where $K \leq N$. For each $i \in [N]$, let $z_i = j$, where $j \in [K]$ is the unique index that satisfies $\tilde{\theta}_i = \theta_j$. We have essentially identified K clusters, and the association of N observations to them, by joining together observations generated by the same parameter. Rewriting the predictive distribution above in terms of these cluster assignment variables:

$$\mathbb{P}\left(z' = j \,\middle|\, z_1, \ldots, z_N\right) = \frac{1}{\alpha + N}\left[\alpha\, \mathbb{I}[j = K+1] + \sum_{k=1}^{K} N_k\, \mathbb{I}[j = k]\right] \tag{2.32}$$

$\mathbb{I}[\cdot]$ is the indicator function, and N_k is the number of points already assigned to cluster k. The first term within the brackets corresponds to assigning o' to a *new* cluster, with cluster index $(K+1)$

and new parameter θ_{K+1}. Equivalently, in the previous section's notation:

$$\mathbb{P}\left(z' = j \,\middle|\, O, Z\right) = \begin{cases} \frac{N_j}{\alpha+N}\,, & \text{cluster } j \text{ exists} \\ \frac{\alpha}{\alpha+N}\,, & j \text{ is a new cluster} \end{cases} \tag{2.33}$$

This is often referred to as the Chinese restaurant process (CRP) (Aldous, 1985), the metaphor being that it is a stochastic process for assigning an infinite stream of restaurant customers to tables. One feature of this process is that the probability of joining a table is proportional to the number of customers already at the table, leading to a 'rich gets richer' phenomenon. The same process in the space of parameters $\tilde{\theta}$, if sampled according to Equation 2.31, is known as the Pólya urn model or the Blackwell-MacQueen sampling scheme (Blackwell and MacQueen, 1973).

In the finite case, where assignments can only be made to existing clusters, recall that:

$$\mathbb{P}_{\text{finite}}\left(z' = j \,\middle|\, O, Z\right) = \frac{\frac{\alpha}{K} + N_j}{\alpha + N} \quad \text{(from Equation 2.25)} \tag{2.34}$$

As $K \to \infty$, this probability approaches the one in the existing-cluster case of the CRP, which is another way in which DPMMs are a natural generalization of finite mixture models. Inference in DPMMs can be performed in a similar fashion, where the CRP now appears in the probability of cluster assignments (Equation 2.23). Given cluster assignments, the rest of the inference process over parameters Θ is basically the same, except that the number of elements in Θ grows.

Numerous tutorials and surveys have been written on Bayesian nonparametric models; useful ones include those by Sudderth (2006), Orbanz and Teh (2010), and Ghahramani (2013). DPs and DPMMs are also discussed in further detail in Teh (2010) and Gershman and Blei (2012).

2.2.3 Dependent Dirichlet process mixture models (DDPMM)

The DPMM's popularity stems from its simplicity and elegance. However, one major limitation is that clusters cannot change over time, a consequence of the fact that observations are assumed to be fully exchangeable. This assumption is violated for problems like target tracking and world modeling, where the observed entities change over time and space. Various generalizations of the DP that model temporal dynamics have thus been proposed (e.g., Zhu et al., 2005; Ahmed and Xing, 2008; Huang et al., 2015) and have been applied to multi-object tracking problems (Campbell et al., 2013; Neiswanger et al., 2014; Luo et al., 2015).

Many of these generalizations belong to a broad class of stochastic models known as dependent Dirichlet processes (DDP) (MacEachern, 1999, 2000). We will consider a theoretically-appealing instance of the DDP, based on a recently-proposed Poisson-process construction (Lin et al., 2010; Lin, 2012). This construction subsumes a number of existing algorithmically-motivated DP generalizations. Additionally, Lin's construction has the nice property that at each time slice, the prior over clusters is marginally a DP. Given a DP prior at time t, the construction specifies a dependent prior at time $t + 1$ (or another future time), which is shown to also be a DP. The construction

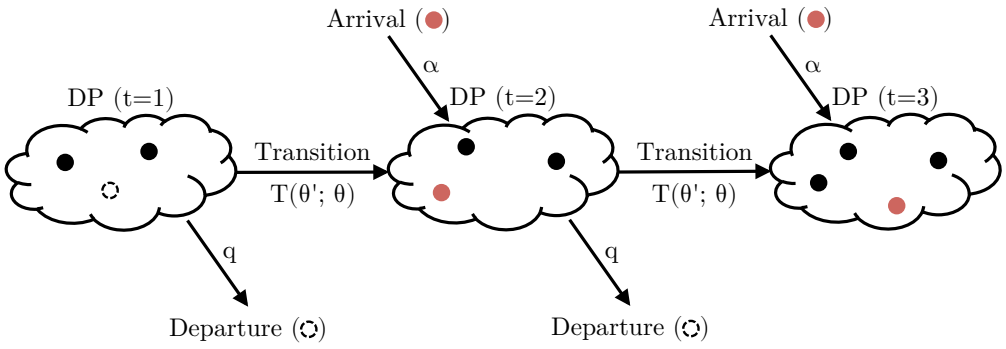


Figure 2-2: An illustration of the dependent Dirichlet process (DDP) construction by Lin et al. (2010). Each 'cloud' represents a DP sample, which can be expressed as a collection of weighted atoms in parameter space Ω. To construct a dependent sequence of DPs, several transition steps are applied to the atoms: subsampling (departure), point transition (transition), and superposition (arrival). If these three steps are performed appropriately (see text for details), then Lin et al. (2010) showed that the resulting set of atoms also forms a DP. Repeating this procedure gives a Markov chain of DPs, which collectively forms a DDP.

therefore generates a Markov chain of DPs over time (see Figure 2-2).

Lin et al. (2010) exploited the fact that there exists a one-to-one correspondence between DPs over parameter space Ω and spatial Poisson processes in the product space $\Omega \times \mathbb{R}_+$. This means that an underlying Poisson process can be extracted from any DP, and vice versa. By considering transitions on the underlying Poisson processes, and restricting to transition steps where the Poisson process remains closed under transition (more fundamentally, by preserving complete randomness), we obtain a new spatial Poisson process at the next time step, which can be converted to a new DP.

According to the stick-breaking construction of the DP (Sethuraman, 1994), if $G^t \sim \text{DP}$, then it can be expressed as infinite sum of weighted atoms: $G^t = \sum_{i=1}^{\infty} w_i \delta_{\theta_i}$, where $w_i \in \mathbb{R}_+$, and $\theta_i \in \Omega$. Then the following DP-preserving transition steps are applied in order:

- Subsampling (removal/departure): Let $q : \Omega \to [0, 1]$ be a parameter-dependent survival rate, i.e., $q(\theta)$ specifies how likely some θ in the current time step survives in the next time step. For each atom θ_i, draw $b_i \sim \text{Ber}\left(q(\theta_i)\right)$, and retain atoms with $b_i = 1$. Renormalizing the weights on the retained atoms gives a new DP sample $G' = \sum_{i:b_i=1} w'_i \delta_{\theta_i}$ (where $\sum_{i:b_i=1} w'_i = 1$).

- Point transition (movement/transition): Let $T\left(\cdot\,;\theta\right) : \Omega \to \mathbb{R}_+$ be a parameter-dependent transition function, i.e., $T\left(\theta'\,;\theta\right)$ specifies how likely some θ in the current time step moves to θ' in the next time step, given that it survives. For each atom θ_i, draw $\theta'_i \sim T\left(\cdot\,;\theta\right)$. Then $G'' = \sum_{i:b_i=1} w'_i \delta_{\theta'_i}$ is a sample from a new DP.

- Superposition (addition/arrival): Let $\Delta = \sum_j \varpi_j \delta_{\vartheta_j}$ be a new independent DP sample, and let $(c, d) \sim \text{Dir}\left(\alpha'', \alpha\right)$, where α'' and α are the concentration parameters of the DPs corresponding to G'' and Δ respectively. Then the random convex combination $G^{t+1} = cG'' + d\Delta$ is a sample from a new DP, and can be used as the prior for the next time step.

The upshot of this DDP construction is that, if we marginalize out the DP prior, we get the following prior for θ^{t+1}, given the parameters from the previous time $\Theta^t = \{\theta^{kt}\}$:

$$\theta^{t+1} \mid \Theta^t \propto \alpha H\left(\theta^{t+1}\right) + \sum_k q(\theta^{kt})\, N^{k,\leq t}\, T\left(\theta^{t+1}\,;\,\theta^{kt}\right) \tag{2.35}$$

The first term is for new atoms, drawn from a DP with base distribution $H(\theta)$ and concentration parameter α.[2] The second term corresponds corresponds to existing atoms that have undergone subsampling and transition steps; these steps affect the assignment probability, as indicated by the presence of q and T. Additionally, $N^{k,\leq t}$ is the number of points that have been assigned to cluster k, for all time steps up to t. This term is similar to N_j in the previous section on the DP. Notice that if $q \equiv 1$ and $T\left(\cdot\,;\,\theta^{kt}\right) = \delta_{\theta^{kt}}$, then we exactly get back the predictive distribution in the DP.

Since $\theta^{t+1} \sim G^{t+1}$, and G^{t+1} is sampled from a DP, we can find the predictive distribution of θ^{t+1}, conditioning also on parameters Θ^{t+1} that have been observed/instantiated at time $(t+1)$:

$$\begin{aligned}\theta^{t+1} \mid \Theta^t, \Theta^{t+1} \propto\ \alpha H\left(\theta^{t+1}\right) &+ \sum_{k:N^{k,t+1}>0} N^{k,\leq t+1}\, \delta_{\theta^{k,t+1}}(\theta^{t+1}) \\ &+ \sum_{k:N^{k,t+1}=0} q(\theta^{kt})\, N^{k,\leq t}\, T\left(\theta^{t+1}\,;\,\theta^{kt}\right)\end{aligned} \tag{2.36}$$

Compared to the DP predictive distribution in Equation 2.31, the summation over existing atoms Θ^t is split into two cases. The first summation above is over atom indices that already exist in the current time $(t+1)$ with parameter value $\theta^{k,t+1}$. Similar to the DP, there is some chance of sampling the same parameter again (i.e., assign to the same cluster), with probability proportional to the historical size of the cluster. The second summation corresponds to a new case: these are atom indices that have not been observed/instantiated at time $(t+1)$, but have existed in the past. If so, the previous-time atom may be 'revived', subject to the event that it survives (with probability $q(\theta^{kt})$). Also, the parameter value is subject to a transition step.

In general, some atoms may not be observed for several time steps, but still affect the prior (with decayed weight and dispersed parameter values). In the existing-cluster case, the subsampling and transition steps must be accumulated over multiple time steps, starting from time $\tau = \tau^{kt}$ when the cluster k was last observed. The general form of the prior on θ^t is:

$$\theta^t \mid \Theta^{\leq t} \propto\ \alpha H\left(\theta^t\right) + \sum_{k:N^{kt}>0} N^{k,\leq t}\, \delta_{\theta^{kt}}(\theta^t) + \sum_{k:N^{kt}=0} q^{kt}\, N^{k,\leq \tau}\, T\left(\theta^t\,;\,\theta^{k\tau}\right)\ , \tag{2.37}$$

$$\text{where } q^{kt} \triangleq \left[q(\theta^{k\tau})\right]^{t-\tau}\ ;\ T\left(\theta^t\,;\,\theta^{k\tau}\right) \triangleq \int\cdots\int \prod_{t'=\tau+1}^{t} T\left(\theta^{t'}\,;\,\theta^{t'-1}\right) \mathrm{d}\theta^{\tau+1}\cdots\mathrm{d}\theta^{t-1}$$

This result will be used in Chapter 4 to derive the DDPMM Gibbs sampling inference algorithm.

[2] Technically, α includes both the innovation process from the superposition step, as well as a subsampled and transitioned version of innovation processes from previous times; see Lin (2012). Also, H can be time-dependent.

2.3 Data Fusion

Due to uncertainty in our perception of the world, in order to improve our estimates and decrease uncertainty, we need to aggregate information from multiple pieces of information, and possibly multiple sources. For example, random noise can be mitigated by repeated observations, a limited field of view can be supplemented by additional judiciously-placed sensors, and information that cannot be captured by one type of sensor may be captured by another type (e.g., sensors for different parts of the electromagnetic spectrum). Integrating such information is the process of data fusion.

Data fusion is a fundamental part of information processing; arguably, this entire dissertation is on various aspects of data fusion. Since it is such a broad and basic area, I will not attempt to cover the background here; the interested reader is invited to consult collections such as Liggins et al. (2008) and surveys such as Khaleghi et al. (2013).

Although I only consider probabilistic data fusion, much data fusion work has been done on other representation of uncertainty, such as fuzzy logic, Dempster-Shafer evidence theory, and random set methods (Section 2.1.3). For some applications, these representations have advantages over probabilistic ones; for example, they can encode ambiguous human operator statements such as "near" without having to assign or compute probabilities over the space of distances. In particular, Mahler (2007) has pushed for a unified data fusion theory based on random finite sets, and has shown in great detail that RFSs can encompass all of the above popular uncertainty representations. Whether RFS methods becomes widely adopted for data fusion remains to be seen.

Shifting back to the world of probabilistic data fusion, Bayesian inference is essentially the main, and only, workhorse (Durrant-Whyte and Henderson, 2008). Where it becomes interesting is when the diversity in information sources causes their data to be *mismatched*. In Chapter 5, I will attempt to fuse information from multiple state representations and filters that operate in very different spaces. This is different in nature compared to Chapters 3 and 4, where the same form of measurements are fused across space and time. Chapter 5 is also different in that it does not require much technical background (this background chapter was mostly written for Chapters 3 and 4); all we need for Chapter 5 is Bayes' rule, and an open mind.

Chapter 3

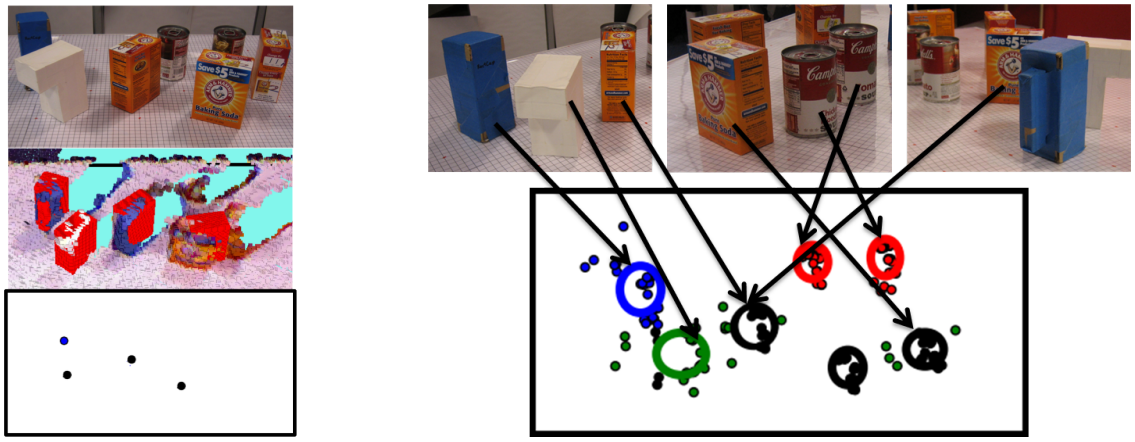
Data Association for Semantic World Modeling from Partial Views

Much of the everyday human physical environment is made up of coherent physical objects. Environmental dynamics are well described in terms of the effects of actions on those objects. Perceptual systems are able to report detections of objects with type, location, color, and other properties. Humans naturally designate both goals and prior information in terms of objects. Thus, it is appropriate for robots to construct 'mental models' of their environment that are structured around objects, their properties, and their relations to one another.

In this chapter, I define a semantic world model to be a set of objects with associated attributes and relations. To illustrate this concept concretely, consider the following tasks, along with objects and attributes that are potentially relevant:

- Cooking eggs on a pan: Objects — Eggs, pan, stove, etc.
  Attributes — *CookedTime*, *StoveSetting*, *EggPositionRelativeToPan*
- Finding chairs for guests: Objects — Furniture, people
  Attributes — *IsChair*, *Sittable*, *Movable*, *Location*, *SittingOn(Person, Furniture)*
- Rearranging objects on a table: Objects — Items on table
  Attributes — *Shape*, *Type*, *RelativePositionAndOrientation*, *GraspPoints*

A common theme underlying these tasks, and many others, is that successful planning and execution hinges on good world-state estimation and monitoring. Dynamic attributes listed above also highlight why object-based representations are uniquely suitable for dynamic tasks: transition dynamics tends to operate on the level of objects. For example, it is much more natural to express and reason about eggs that are being cooked, as opposed to points in a point cloud or cells in an occupancy grid that are 'cooked'. Although we focus on the static case in this chapter, our ultimate goal is to provide a framework for estimating and monitoring large semantic world models involving objects and attributes that change over time as a result of physical processes as well as actions by the robot and other agents. I will generalize the model to semi-static environments in Chapter 4.



(a) Single viewpoint **(b)** Aggregation of object detections from multiple viewpoints

Figure 3-1: (a) Given a tabletop scene (top), we want to estimate the types and poses of objects in the scene using a black-box object detector. From a single RGB-D image, however, objects may be occluded or erroneously classified. In the rendered image (middle; detections superimposed in red), three objects are missing due to occlusion, and the bottom two objects have been misidentified. The semantic attributes that result in our representation are very sparse (bottom; dot location is measured 2-D pose, color represents type). A single viewpoint is typically insufficient to identify all objects in a scene correctly.
(b) Aggregation of measurements from many different viewpoints (top) is therefore needed to construct good estimates. However, this introduces data association issues of the type addressed in this work, especially when multiple instances of the same object type are present. From all the object detection data, as shown (bottom) by dots (each dot is one detection), our goal is to estimate the object types and poses in the scene (shown as thick ellipses centered around location estimate; color represents type, ellipse size reflects uncertainty). The estimate above identifies all types correctly with minimal error in pose.

In this chapter, I address the problem of constructing world models from semantic perception modules that provide noisy observations of attributes. For concreteness, Figure 4-1 depicts an application of our methods; here the world model consists of objects' types and poses, and attribute measurements are outputs from a black-box object detector running continuously on sensed RGB-D images. Due to noise, occlusion, and sensors' limited field of view, observations from multiple viewpoints will typically be necessary to produce a confident world model. Because attribute detections are sparse, noisy, and inherently ambiguous, where it is unclear which attribute measurements were produced by the same object across different views, *data association* issues become critical. This is the greatest challenge; if the measurement-object correspondences were known, the resulting object-attribute posterior distributions would be efficiently computable.

I begin by stating a formal model for a simplified 1-D version of the world-model estimation problem in Section 3.2, and then review a classic solution approach based on tracking in Section 3.3. The main contribution of this chapter is the development of several novel clustering-based data association approaches, described in Sections 3.4 and 3.5. Application of the semantic world-modeling framework to object type-and-pose estimation is then demonstrated in Section 3.6, where I present experimental results using data collected with a Kinect sensor on a mobile robot.

3.1 Related Work

The work in this chapter lies in the intersection of semantic perception, world modeling, and data association. The first two were discussed in Chapter 1, and the latter was covered in Chapter 2.

In the context of previous work, I view my approach as building on the semantic world modeling problem formulation of Elfring et al. (2013) (which used multiple hypothesis tracking (MHT) to perform data association) and the data association techniques (MCMCDA) of Oh et al. (2009). As argued by Oh et al. (2009), MHT has various drawbacks, which are directly inherited by the approach of Elfring et al. (2013). However, instead of directly applying MCMCDA to world modeling, I will introduce more domain assumptions to make inference more efficient.

Unlike target tracking, for which most data association algorithms are designed, semantic world modeling has three distinguishing domain characteristics:

- Objects can have attributes besides location, and hence are distinguishable from each other in general (which likely makes data association easier). Some data association methods can be readily generalized to this case as was done by Elfring et al. (2013) for the MHT.
- Only a small region of the world is visible from any viewpoint. Most data association methods operate in regimes where all targets are sensed (possibly with noise/failure) at each time point.
- Most object states do not change over short periods of time.

With respect to the first point, Dellaert (2001) also recognized the utility of considering attributes in data association problems. When occlusions and clutter are present, correspondences are frequently ambiguous, and incorporating more information can help separate the targets of correspondence. Dellaert (2001) considered this idea in the context of the structure-from-motion problem, proposing that image feature appearances should be considered in addition to their measured locations, in order to better distinguish different features between images. My approach shares many resemblances to this line of work due to the use of attributes and MCMC-based inference.

In light of the final point, we study the semantic world modeling problem under the stringent assumption that the world is static, i.e., object states do not change.[1] This does not trivialize the data association problem, since it is still necessary to determine measurement-to-object correspondences (and is exacerbated by the limited field of view). However, target-tracking algorithms no longer seem most appropriate, since time is no longer an essential dimension. Instead, the problem becomes more akin to *clustering*, where objects are represented by points in the joint attribute (product) space, and measurements form clusters around these points.

A useful model for performing clustering with an unbounded number of clusters is the Dirichlet process mixture model (DPMM) (Antoniak, 1974; Neal, 2000), a Bayesian nonparametric approach that can be viewed as an elegant extension to finite mixture models (see Section 2.2 for details). I

[1] Over long periods of time, this assumption is clearly unrealistic, but is beyond the scope of this chapter. A naïve solution is to refresh the world model using a short window of measurements prior to each query, assuming that the world has not changed during that window.

apply this clustering method to world modeling in Section 3.4 and derive a Gibbs sampling algorithm to perform inference. The sampling candidate proposals in this algorithm can be viewed as a subset of those considered by Oh et al. (2009). However, clustering ignores the crucial cannot-link constraint (CLC) in data association; more details will be given in Section 3.5, where I introduce modifications and approximations to address this issue.

3.2 The 1-D Colored-Lights Domain

For clarity of explanation I begin by introducing a model of minimal complexity, involving objects with 1-D locations and a single attribute (color). Despite this simplification, the fundamental issues in data association are captured in the model described in this section. Generalizing to higher dimensions and more attributes is relatively straightforward; in Section 3.6, I generalize to 3-D locations and use object types as an attribute in our semantic world modeling application.

The world consists of an unknown number (K) of stationary lights. Each light is characterized by its color c_k and its location $l_k \in \mathbb{R}$, both of which do not change over time. A finite universe of colors of size C is assumed. A robot moves along this 1-D world, occasionally gathering partial views of the world with known fields of view $[a^v, b^v] \subset \mathbb{R}$. Within each view, M^v lights of various colors and locations are observed, denoted by $o_m^v \in [C] \triangleq \{1, \ldots, C\}$ and $x_m^v \in \mathbb{R}$ respectively. These (o_m^v, x_m^v) pairs may be noisy (in both color and location) or spurious (false positive – FP) measurements of the true lights. Also, a light may sometimes fail to be perceived (false negative – FN). Given these measurements, the goal is to determine the posterior distribution over configurations (number, colors, and locations) of lights in the explored region of the world.

I assume the following form of noise models. For color observations, for each color c, there is a known discrete distribution $\phi^c \in \Delta^C$ (estimable from perception apparatus statistics) specifying the probability of color observations:

$$\phi_i^c = \begin{cases} \mathbb{P}(\text{no observation for light with color } c) \,, & i = 0 \\ \mathbb{P}(\text{color } i \text{ observed for light with color } c) \,, & i \in [C] \end{cases} \tag{3.1}$$

A similar distribution ϕ^0 specifies the probability of observing each color given that the observation was a false positive. False positives are assumed to occur in a proportion p_{FP} of object detections. Each view may have multiple detections and hence multiple false positives. For location observations, if the observation corresponds to an actual light, then the observed location is assumed to be Gaussian-distributed, centered on the actual location. The variance is *not* assumed known and will be estimated for each light from measurement data. For false positives, the location is assumed to be uniformly distributed over the field of view (Unif$[a^v, b^v]$).

Next, I present the core problem of this domain. Given sets of color-location detections from a sequence of views, $\{\{(o_m^v, x_m^v)\}_{m=1}^{M^v}\}_{v=1}^{V}$, we want to infer the posterior distribution on the configuration of lights $\{(c_k, l_k)\}_{k=1}^{K}$, where K is unknown as well. If we knew, for each light, which subset of the measurements were generated from that light, then we would get K decoupled estimation

problems (assuming lights are independent from each other). With suitable priors, these single-light estimation problems admit efficient solutions; details can be found in the Appendix.

The issue is that these associations are unknown. Therefore, we must reason over the *space* of possible data associations. For each observation, let z_m^v be the index of the light that the observation corresponds to (ranging in $[K]$ for a configuration with K lights), or 0 if the observation is a false positive. z_m^v is the latent association for measurement (o_m^v, x_m^v). Let $\mathbf{z}^v$ be the concatenated length-M^v vector of all z_m^v variables in view v, and let $\{\mathbf{z}^v\}$ be the collection of all correspondence vectors from the V views. We then aggregate estimates over all latent associations (some indices have been dropped to reduce clutter, if clear from context; please refer to the previous paragraph for indices):

$$\mathbb{P}\Big(\{(c,l)\} \;\Big|\; \{\{(o,x)\}\}\Big) = \sum_{\{\mathbf{z}^v\}} \mathbb{P}\Big(\{(c,l)\} \;\Big|\; \{\mathbf{z}^v\}, \{\{(o,x)\}\}\Big)\, \mathbb{P}\Big(\{\mathbf{z}^v\} \;\Big|\; \{\{(o,x)\}\}\Big) \tag{3.2}$$

The first term is given by the decoupled estimation problems mentioned above, and results in a closed-form posterior distribution given in the Appendix (Section 3.A). The desired posterior distribution on the left is therefore, in exact form, a mixture over the closed-form posteriors. The problem is that the number of mixture components is exponential in M^v and V, one for each full association $\{\mathbf{z}^v\}$, so maintaining the full posterior distribution is intractable. Finding tractable approximations to this light-configuration posterior distribution is the subject of Sections 3.3–3.5.

3.3 A Tracking-Based Approach

If we consider the lights to be stationary targets and the views to be a temporal sequence, a target-tracking approach can be used. Tracking simultaneously solves the data association (measurement correspondence) and target parameter estimation (light colors and locations) problems. As discussed in Section 2.1, a wide variety of tracking algorithms exist, and in particular multiple hypothesis tracking (MHT) (Reid, 1979) has already been adopted by Elfring et al. (2013) on the problem of semantic world modeling. We provide a gist of the MHT applied to world modeling, and discuss a problematic issue below; readers are referred to Elfring et al. (2013) for details.

The MHT algorithm maintains, at every timestep (view) v, a distribution over all possible associations of measurements to targets up to v. At each view, MHT therefore needs to propagate *each* previous hypothesis forward with *each* possible association in view v. One way to consider this is as a tree, where nodes of depth v are associations up to view v, and a distribution is maintained on the leaves. Each view introduces a new layer of nodes, where the branching factor is the number of valid associations in that view. Without loss of generality, assume that the views are in chronological order. The distribution over associations up to view v is:

$$\begin{aligned}\mathbb{P}\Big(\{\mathbf{z}\}^{\leq v} \;\Big|\; \{\{(o,x)\}\}^{\leq v}\Big) &= \mathbb{P}\Big(\mathbf{z}^v \;\Big|\; \{\mathbf{z}\}^{<v}, \{\{(o,x)\}\}^{\leq v}\Big)\, \mathbb{P}\Big(\{\mathbf{z}\}^{<v} \;\Big|\; \{\{(o,x)\}\}^{<v}\Big) \\ &\propto \mathbb{P}\Big(\{(o^v,x^v)\} \;\Big|\; \mathbf{z}^v, \{\mathbf{z}\}^{<v}, \{\{(o,x)\}\}^{<v}\Big)\, \mathbb{P}\Big(\mathbf{z}^v \;\Big|\; \{\mathbf{z}\}^{<v}, \{\{(o,x)\}\}^{<v}\Big)\, \mathbb{P}\Big(\{\mathbf{z}\}^{<v} \;\Big|\; \{\{(o,x)\}\}^{<v}\Big)\end{aligned} \tag{3.3}$$

where superscript "v" indicates variables at view v only, "$\leq v$" for everything up to view v, and "$< v$" for everything up to the previous view (excluding v). The first term is the likelihood of the current view's observations, the second is the prior on the current view's correspondences given previously identified targets, and the final term is the filter's distribution from the previous views.

The likelihood term for view v follows mostly from the derivation in the Appendix. The observations are independent given the view's correspondence vector $\mathbf{z}^v$, and the likelihood is a product of M^v of the following terms:

$$\mathbb{P}\Big(o_m^v, x_m^v \,\Big|\, z_m^v = k, \{\mathbf{z}\}^{<v}, \{\{(o,x)\}\}^{<v}\Big) = \begin{cases} \frac{\phi_o^0}{b^v - a^v}, & k = 0 \\ \mathbb{P}\Big(o_m^v \,\Big|\, \{\{o\}\}_{z=k}^{<v}\Big)\, \mathbb{P}\Big(x_m^v \,\Big|\, \{\{x\}\}_{z=k}^{<v}\Big), & k \neq 0 \end{cases} \tag{3.4}$$

where the "$z = k$" subscript refers to observations (from previous time steps in this case) that have been assigned to the same light, as indicated by the correspondence vectors $\{\mathbf{z}\}^{<v}$. Observations corresponding to other lights are ignored because lights are assumed to be independent. The two probability terms can be found from the posterior predictive distribution (Equations 3.24, 3.28 respectively). For new targets (where k does not index an existing target), the conditioning set of previous observations will be empty, but can be likewise handled by the predictive distributions. The false positive probability ($k = 0$) follows from the observation model (Equation 3.1).

The prior on the current view's correspondences, the second term in Equation 3.3, is due to Reid (1979). Assume we know which of the existing targets are within the current field of view based on the hypothesis on previous views (this can be found by gating). Denote the indices of these targets as the size-K^v set $\{k\}^v$. Another plausible assumption used in the tracking literature, due to sensor characteristics, is that in a single view, each target can generate at most one non-spurious measurement. I will refer to this as the *cannot-link constraint* (CLC), because multiple measurements within the same view cannot be 'linked' to the same target.

I now define validity of correspondence vectors $\mathbf{z}^v$. Recall that in this length-M^v vector, the m'th entry z_m^v is the (positive integer) target index to which (o_m^v, x_m^v) correspond, or 0 for a false positive. First, an entry in $\mathbf{z}^v$ must either be 0, a target index in $\{k\}^v$, or a new (non-existing) index; otherwise, it corresponds to an out-of-range target. Second, by the CLC assumption, no entry may be repeated in $\mathbf{z}^v$, apart from 0 for false positives. A correspondence $\mathbf{z}^v$ is valid if and only if it satisfies both conditions.

The following quantities can be found directly from $\mathbf{z}^v$:

$$\begin{aligned} n_0 &\triangleq \text{Number of false positives (0 entries)} \\ n_\infty &\triangleq \text{Number of new targets (non-existing indices)} \\ \delta_k &\triangleq \mathbb{I}\big[\text{Target } k \text{ is detected } (\exists m.\, z_m^v = k)\big], k \in \{k\}^v \\ n_1 &\triangleq \text{Number of matched targets} = M^v - n_0 - n_\infty = \sum_k \delta_k \text{ (by CLC)} \end{aligned} \tag{3.5}$$

where $\mathbb{I}[\cdot]$ is the indicator function. Then we can split $\mathbb{P}\big(\mathbf{z}^v \,\big|\, \{\mathbf{z}\}^{<v}, \{\{(o,x)\}\}^{<v}\big)$ by conditioning on the above quantities, which are deterministic functions of $\mathbf{z}^v$:[2]

$$\mathbb{P}\big(\mathbf{z}^v\big) = \mathbb{P}\big(\mathbf{z}^v, n_0, n_\infty, n_1, \{\delta_k\}\big) = \mathbb{P}\big(\mathbf{z}^v \,\big|\, n_0, n_\infty, n_1, \{\delta_k\}\big)\, \mathbb{P}\big(n_0, n_\infty, n_1, \{\delta_k\}\big) \tag{3.6}$$

By the assumed model characteristics, the second term is:

$$\mathbb{P}\big(n_0, n_\infty, n_1, \{\delta_k\}\big) = \text{Binomial}\big(n_0\,;\, M^v, p_{\text{FP}}\big)\, \mathbb{P}\big(n_\infty\,;\, M^v\big)\, \mathbb{P}\big(\{\delta_k\}\big) \tag{3.7}$$

$$\mathbb{P}\big(\{\delta_k\}\big) = \prod_{k \in \{k\}^v} \big[p_{\text{D}}(k)\big]^{\delta_k} \big[1 - p_{\text{D}}(k)\big]^{1-\delta_k} \tag{3.8}$$

where $p_{\text{D}}(k)$ is the (target-specific) detection probability defined in Equation 3.25 in the Appendix. The number of new targets n_∞ is typically Poisson-distributed.

Determining the correspondence given the quantities above involves assigning $\mathbf{z}_m^v$ indices to the three groups of entries (of sizes n_0, n_∞, and n_1) and matching a size-n_1 subset of $\{k\}^v$ (as indicated by $\{\delta_k\}$) to the indices in the final group. A common assumption is that all assignments and matches of indices are equally likely, so the first term in Equation 3.6 is the reciprocal of the number of valid correspondence vectors (given n_0, n_∞, n_1, and $\{\delta_k\}$), given by:

$$n_{\text{valid}}\big(n_0, n_\infty, n_1, \{\delta_k\}\big) = \binom{M^v}{n_0, n_\infty, n_1}\, n_1! = \frac{M^v!}{n_0! n_\infty!} \tag{3.9}$$

Combining Equations 3.4–3.9 gives the necessary expressions used in the MHT filter (Equation 3.3).

The expression for n_{valid}, which is related to the branching factor in the tree of associations that the MHT considers, highlights the complexity of this approach. To obtain the total number of valid associations, we need to also consider all possible settings of n_0, n_∞, n_1, and $\{\delta_k\}$:

$$n_{\text{total}} = \sum_{n_0=0}^{M^v} \sum_{n_\infty=0}^{(M^v - n_0)} \binom{K^v}{n_1}\, n_{\text{valid}}\big(n_0, n_\infty, n_1, \{\delta_k\}\big) \tag{3.10}$$

Even with 4 measurements and 3 within-range targets, the branching factor is 304, so considering all hypotheses over many views is clearly intractable. Many hypothesis-pruning strategies have been devised (e.g., Kurien (1990); Cox and Hingorani (1996)), the simplest of which include keeping the best hypotheses or hypotheses with probability above a certain threshold. More complex strategies to combine similar tracks and reduce the branching factor have also been considered. In the experiments of Section 3.6 I simply keep hypotheses with probability above a threshold of 0.01. As I will demonstrate in the experiments, an MHT filter using this aggressive pruning strategy can potentially cause irreversible association errors and make incorrect conclusions.

[2] The probabilities implicitly depend on previous correspondences $\{\mathbf{z}\}^{<v}$ and observations $\{\{(o,x)\}\}^{<v}$, as shown in the second term of Equation 3.3, via the targets in view $\{k\}^v$ and their detection probabilities in Equation 3.8.

3.4 A Clustering-Based Approach

If we consider all the measurements together and disregard their temporal relationship (static world assumption), we expect the measurements to form clusters in the product space of colors and locations ($[T] \times \mathbb{R}$), allowing us to derive estimates of the number of lights and their parameters. In probabilistic terms, the measurements are generated by a mixture model, where each mixture component is parameterized by the unknown parameters of a light. Since the number of lights in the world is unknown, we also do not want to limit the number of mixture components a priori.

As mentioned in Section 3.1, the Dirichlet process mixture model (DPMM) supports an unbounded number of mixture components (see Section 2.2 for details). I now derive the DPMM model specifics and inference procedure for the colored-lights domain. A few more assumptions need to be made and parameters defined. Our model assumes that the cluster parameter distribution G is drawn from a DP prior $\mathrm{DP}(\alpha, H)$, where H is the base distribution and α is the concentration hyperparameter (controlling the similarity of G and H, and also indirectly the number of clusters). H acts as a 'template' for the DP, and is hence also a distribution over the space of cluster parameters. I set it to be the product distribution of π, the prior on colors c, and a normal-gamma distribution over the location l and its observation precision τ (see the Appendix for details on this latter distribution, as well as an interpretation of the subscripted hyperparameters):

$$H(c, l, \tau) \triangleq \pi_c \,\mathrm{NormalGamma}\,(l, \tau\,;\, \lambda_0, \nu_0, \alpha_0, \beta_0) \tag{3.11}$$

To accommodate false positives, which occur with probability p_{FP}, we scale G from the DP prior by a factor of $(1 - p_{\mathrm{FP}})$ for true positives, and let the remaining probability mass correspond to a parameter-less cluster to which all false positives are assigned.

As discussed in Section 2.2.2, we can use the stick-breaking construction of Sethuraman (1994) to express G explicitly as a sum of weighted atoms. In particular, generate:

$$\{\pi_k\} \sim \text{Griffiths-Engen-McCloskey}(\alpha)\;;\; \{(c_k, l_k, \tau_k)\} \sim H \tag{3.12}$$

Once $\{\pi_k\}$ and $\{(c_k, l_k, \tau_k)\}$ are drawn from the DP prior, the rest of the generative process is:

$$\theta_k = \begin{cases} p_{\mathrm{FP}}\;, & k = 0 \\ (1 - p_{\mathrm{FP}})\,\beta_k\;, & k \neq 0 \end{cases} \qquad \text{Cluster proportions (with FPs)} \tag{3.13}$$

$$z_m^v \sim \theta\;; \qquad m \in [M^v]\;,\; v \in [V] \qquad \text{Cluster assignment (for each obs.)}$$

$$o_m^v \sim \begin{cases} \phi^0\;, & z_m^v = 0 \\ \phi^{c_z}\;, & z_m^v \neq 0 \end{cases} \qquad \text{Color observation}$$

$$x_m^v \sim \begin{cases} \mathrm{Unif}[a^v, b^v]\;, & z_m^v = 0 \\ \mathcal{N}\left(l_z, \tau_z^{-1}\right)\;, & z_m^v \neq 0 \end{cases} \qquad \text{Location observation}$$

Despite being a nice theoretical tool, an infinite collection of sticks cannot be directly handled computationally. The most straightforward way to perform inference in a DPMM is by Gibbs sampling. In particular, I derive a collapsed Gibbs sampler for the cluster correspondence variables $\{\{z_m^v\}\}$ and integrate out the other latent variables, (c, l, τ) and θ. In Gibbs sampling, we iteratively sample from the conditional distribution of each z_m^v, given all other correspondence variables (which I will denote by $\{\{z\}\}^{-vm}$). By Bayes' rule:

$$\begin{aligned}
&\mathbb{P}\Big(z_m^v = k \,\Big|\, \{\{z\}\}^{-vm}, \{\{(o,x)\}\}\Big) \\
&\quad\propto \mathbb{P}\Big(o_m^v, x_m^v \,\Big|\, z_m^v = k, \{\{z\}\}^{-vm}, \{\{(o,x)\}\}^{-vm}\Big)\, \mathbb{P}\Big(z_m^v = k \,\Big|\, \{\{z\}\}^{-vm}, \{\{(o,x)\}\}^{-vm}\Big) \\
&\quad\propto \mathbb{P}\Big(o_m^v, x_m^v \,\Big|\, \{\{(o,x)\}\}_{z=k}^{-vm}\Big)\, \mathbb{P}\Big(z_m^v = k \,\Big|\, \{\{z\}\}^{-vm}\Big)
\end{aligned} \tag{3.14}$$

In the final line, the first term can be found from the posterior predictive distributions described in the Appendix (Equations 3.24 and 3.28), in a similar fashion to that in the MHT (Equation 3.4). This allows us to collapse the latent cluster parameters (c, l, τ). Note that the observations being conditioned on *exclude* (o_m^v, x_m^v) and depend on the current correspondence variable samples (to determine which observations belong to cluster k).

The second term, the distribution of z_m^v given all other cluster assignments, is given by the Chinese restaurant process (CRP), which is obtained by integrating out the DP prior on θ. Together with our prior on false positives:

$$\mathbb{P}\Big(z_m^v = k \,\Big|\, \{\{z\}\}^{-vm}\Big) = \begin{cases} (1 - p_{\text{FP}})\, \frac{N_k^{-vm}}{\alpha + N^{-vm}}\,, & k \in [K] \ (k \text{ exists}) \\ (1 - p_{\text{FP}})\, \frac{\alpha}{\alpha + N^{-vm}}\,, & k = K + 1 \ (k \text{ new}) \\ p_{\text{FP}}\,, & k = 0 \end{cases} \tag{3.15}$$

where N_k^{-vm} is the number of observations currently assigned to cluster k (excluding (o_m^v, x_m^v)), $N^{-vm} = \sum_{k=1}^{K} N_k^{-vm}$ is the total number of non-false-positive observations across all views, and K is the number of instantiated clusters. This expression illustrates the role of the concentration parameter α from a sampling perspective: larger α leads to more frequent creation of new clusters.

By combining Equations 3.14 and 3.15, we can sample from the conditional distribution of individual correspondences z_m^v. Although the model supports an infinite number of clusters, the modified CRP expression (Equation 3.15) shows that we only need to compute $K' + 2$ values for one sampling step, where K' is the number of existing clusters with $N^{-vm} > 0$, which is finite, since clusters without data are removed. One sampling sweep over all correspondence variables $\{\{z_m^v\}\}$ constitutes one sample from the DPMM. Given the correspondence sample, finding the posterior configuration is simple. Each non-empty cluster corresponds to a light. For each cluster, applying Equations 3.23 and 3.26 from the Appendix to the cluster's associated data provides the posterior distributions on the light's color and location (with observation model precision) respectively. The posterior marginal distribution on the light's location is a t-distribution with parameters given in Equation 3.27.

Input: Observations $\{\{(o_m^v, x_m^v)\}_{m=1}^{M^v}\}_{v=1}^{V}$
Number of samples S
Output: Samples of cluster assignments $\{\{\{z_m^{v\,(s)}\}_{m=1}^{M^v}\}_{v=1}^{V}\}_{s=1}^{S}$

Init. $K := 0$; $z_m^{v\,(0)} := 0$ for all $m \in [M^v], v \in [V]$
for $s := 1$ **to** S; $v := 1$ **to** V; $m := 1$ **to** M^v **do**
  Find cluster predictive distributions and sizes using most-recent samples $\{\{z_m^{v\,(s-1)}\}\}^{-vm}$
  Compute sampling distribution (Equation 3.14) by multiplying Equation 3.4 and Equation 3.15 for each $k \in \{0\} \cup [K+1]$, then normalizing
  Sample $z_m^{v\,(s)}$ from sampling distribution
  if $z_m^{v\,(s)} = K+1$ **then**
    $K := K+1$
  Remove cluster with index $z_m^{v\,(s-1)}$ if it has no other assigned observations

(a) Collapsed Gibbs sampling for DPMM

Input: Observations $\{\{(o_m^v, x_m^v)\}_{m=1}^{M^v}\}_{v=1}^{V}$
Cluster penalty parameter λ
Output: Cluster assignments $\{\{\{z_m^v\}_{m=1}^{M^v}\}_{v=1}^{V}\}$

Init. $K := 1$; $z_m^v := 1$ for all $m \in [M^v], v \in [V]$
repeat
  for $v := 1$ **to** V; $m := 1$ **to** M^v **do**
    $d_m^v(k) := -\log \mathbb{P}\left(o_m^v, x_m^v \mid \{\{(o,x)\}\}_{z=k}\right)$ for each $k \in [K]$ (using Equation 3.4)
    if $\min_k d_m^v(k) > \lambda$ **then**
      $z_m^v := K+1$; $K := K+1$
    else
      $z_m^v := \arg\min_k d_m^v(k)$
until convergence
Sort clusters by size $|\{z_m^v = k\}|$, remove smallest clusters containing a total of a p_{FP}-proportion of all observations, and set associated $z_m^v = 0$

(b) Hard-clustering algorithm for DPMM, inspired by DP-means (Kulis and Jordan, 2012)

Figure 3-2: Inference algorithms for DPMMs, one by sampling, the other by hard clustering.

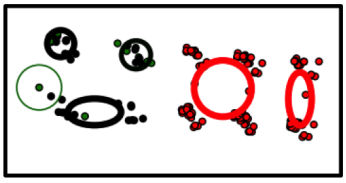
Although Gibbs sampling is a conceptually simple inference method for recovering the posterior distribution in the DPMM, it is relatively inefficient because it requires a substantial number of samples to reach convergence. Kulis and Jordan (2012) recently proposed an elegant hard-clustering method, DP-means, that produces a single clustering assignment. The algorithm is derived from analyzing the small-variance asymptotics of the DPMM Gibbs sampler, and bears great resemblance to k-means. Like k-means, data points are assigned to their closest cluster centers, with the exception that points farther than λ away from all existing clusters are instead assigned to instantiate a new cluster. The process is repeated until convergence, which is guaranteed. The original presentation involved only Gaussian-distributed cluster means. Figure 3-2(b) shows our extension to the algorithm, which handles the discrete color parameter and also false positives. Although this method produces a single assignment instead of a distribution, I will use it in the next section to initialize algorithms that handle more data association constraints.

3.5 Incorporating View-Level Information and Constraints

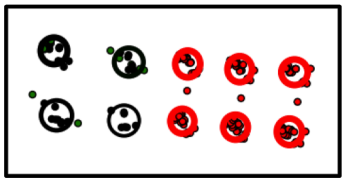
The DPMM-based solution to the colored-lights problem is a straightforward application of the DPMM, but ignores two fundamental pieces of information:

- **Visible region information and false negatives (FN)**: The DPMM does not consider the field of view $[a^v, b^v]$, and hence neither which clusters are visible when a measurement is made. Clusters that are frequently visible but only sporadically detected suggest that there may in fact be no light there, that the detections were errors. Because the DPMM does not consider this, it may posit a cluster for a spurious measurement when its absence in other

(a) Single viewpoint



(b) DPMM sample clustering



(c) DPMM-FullView sample

Figure 3-3: A real-world example demonstrating issues with the DPMM approach from Section 3.4. **(a)** A scene containing many instances of the same object type, viewed from above. The relative proximity of similar objects make them easily confusable for data association. **(b)** The DPMM approach performs particularly poorly in this scenario because it ignores false-negative information and the one-measurement-per-object constraint. One sample from the posterior distribution is shown by the thick ellipses, centered around the mean cluster locations, with sizes depicting posterior variance and colors depicting object type (red = red soup can, black = orange baking-soda box); the small dots show raw measurements. Ignoring FNs causes spurious clusters (e.g., left-most thin green circle) to be more likely, since they are not discounted by absences in other views. Ignoring the CLC assumption causes measurements from similar nearby objects to be aggregated into a single cluster, even if they were observed together in a single view, as was the case for the four soup cans in the middle. **(c)** By taking into account view-level information and constraints, the DPMM-FullView method described in Section 3.5.1 recovers the correct interpretation of the scene.

views would have suggested otherwise. It may also assign a measurement to a cluster that should not be visible from the current view, although this case is less likely to occur.

- **Cannot-link constraint (CLC)**: When two lights of the same color are placed close to each other, they are easily confusable. The only way to distinguish between them is if both are consistently detected together. Then, by the CLC assumption, the two detections cannot be assigned to the same light, so the second detection must come from a second light, or be an FP. With sufficient views, two clusters emerge. Because the DPMM ignores the CLC, it may associate both detections to the same cluster. In fact, the DPMM generally prefers larger clusters (instead of two small ones) due to the 'rich gets richer' phenomenon in the Chinese restaurant process (Equation 3.15).

Figure 3-3 illustrates a real-world example of both issues.

The above issues are consequences of the DPMM's conditional independence assumptions. To see this, consider the concrete example depicted in Figure 3-4, where we wish to sample cluster assignments for an entire view's $M^v = 4$ measurements. The DPMM Gibbs sampler samples the cluster assignment for each measurement *individually*, as shown in Figure 3-4(b). This causes the two right-most measurements to be assigned to the same cluster, a violation of the CLC. The assumption states that *at most one* measurement in a single view can be assigned to each cluster; this view-level constraint cannot be incorporated on the level of individual measurements. Likewise, a false negative only arises if *none* of the measurements in a view are assigned to a cluster within the field of view. To handle these constraints we must couple the measurements and sample their assignments *jointly*.

3.5.1 DPMM-FullView

More formally, instead of sampling a view v's correspondence variables $\{z_m^v\}_{m=1}^{M^v}$ one by one, we consider sampling from the conditional distribution of the *joint* correspondence vector $\mathbf{z}^v$:

$$\mathbb{P}\Big(\mathbf{z}^v \;\Big|\; \{\mathbf{z}\}^{-v}, \{\{(o,x)\}\}\Big) \propto \mathbb{P}\Big(\{(o^v,x^v)\} \;\Big|\; \mathbf{z}^v, \{\mathbf{z}\}^{-v}, \{\{(o,x)\}\}^{-v}\Big)\, \mathbb{P}\Big(\mathbf{z}^v \;\Big|\; \{\mathbf{z}\}^{-v}\Big) \tag{3.16}$$

Like the previous two sections, the first term is an observation likelihood term that factors into a product of M^v terms, each of which is similar to Equation 3.4. The second term is the DP conditional distribution on $\mathbf{z}^v$, and can be found by repeated application of the CRP (Equation 3.15):

$$\begin{aligned}\mathbb{P}_{\text{DPMM}}\Big(\mathbf{z}^v \;\Big|\; \{\mathbf{z}\}^{-v}\Big) &= \mathbb{P}\Big(z_{M^v}^v \;\Big|\; z_{M^v-1}^v, \ldots, z_1^v, \{\mathbf{z}\}^{-v}\Big)\; \cdots\; \mathbb{P}\Big(z_2^v \;\Big|\; z_1^v, \{\mathbf{z}\}^{-v}\Big)\, \mathbb{P}\Big(z_1^v \;\Big|\; \{\mathbf{z}\}^{-v}\Big) \\ &= \frac{p_{\text{FP}}^{n_0}\;(1-p_{\text{FP}})^{(n_1+n_\infty)}\;\alpha^{n_\infty}\;\Big[\prod_{\{m\}_1} N_{z_m^v}^{-v}\Big]}{\prod_{m'=0}^{(n_1+n_\infty)-1}\alpha + N^{-v} + m'}\end{aligned} \tag{3.17}$$

where n_0, n_∞, n_1 are the previously-defined functions of $\mathbf{z}^v$ (Equation 3.5), and $\{m\}_1$ is the set of indices that are matched to existing targets (i.e., $n_1 = |\,\{m\}_1\,|$).

To see the derivation, consider the known values of n_0, n_∞, n_1 given $\mathbf{z}^v$. This means that there must be n_0 entries in $\mathbf{z}^v$ with value 0, n_∞ entries with a new positive value, and n_1 entries with an existing positive value. These three types of entries correspond exactly to the cases for the CRP, hence in Equation 3.17, n_0 of the terms must be p_{FP}, and so on. N^{-v} is the total number of non-view-v, non-FP observations, and $N_{z_m^v}^{-v}$ is the number of observations assigned to the cluster with index equal to the value of z_m^v, excluding view v. The latter type of counts are used in the CRP case when the assignment z_m^v corresponds to an existing cluster index k. In general, $N_k^{-vm} \geq N_k^{-v}$ (former from CRP, latter discussed above), so the expression in Equation 3.17 does not hold in general. However, because of the CLC assumption, no other observation in view v could be assigned to cluster k, so in fact $N_k^{-vm} = N_k^{-v}$, and Equation 3.17 holds in our case.

Equation 3.16 is essentially the product of M^v conditional distributions used in the DPMM, and does not yet achieve our goal of incorporating FNs and the CLC assumption. To use FNs and field-of-view information, we take inspiration from the MHT formulation, and first suppose we knew which K^v of the existing K lights are within the field of view, i.e., $\{k\}^v$ from Section 3.3. This, together with $\mathbf{z}^v$, allows us to determine the detection indicator variables $\{\delta_k\}$ (Equation 3.5) and their probabilities (Equation 3.8). For the CLC, we simply assign zero probability to violating correspondences. We combine the additional information with the DPMM-based conditional distribution above (Equation 3.17) in a conceptually simple fashion:

$$\mathbb{P}_{\text{FullView}}\Big(\mathbf{z}^v \;\Big|\; \mathbf{z}^{-v}, \{k\}^v\Big) \propto \mathbb{P}_{\text{DPMM}}\Big(\mathbf{z}^v \;\Big|\; \mathbf{z}^{-v}\Big)\, \mathbb{P}\Big(\{\delta_k\}\Big)\, \mathbb{I}\Big[\mathbf{z}^v \text{ satisfies CLC}\Big] \tag{3.18}$$

The final term evaluates to 1 if the joint correspondence satisfies the CLC, and 0 otherwise. Hence by construction the correspondence variables sampled from this conditional distribution will incor-

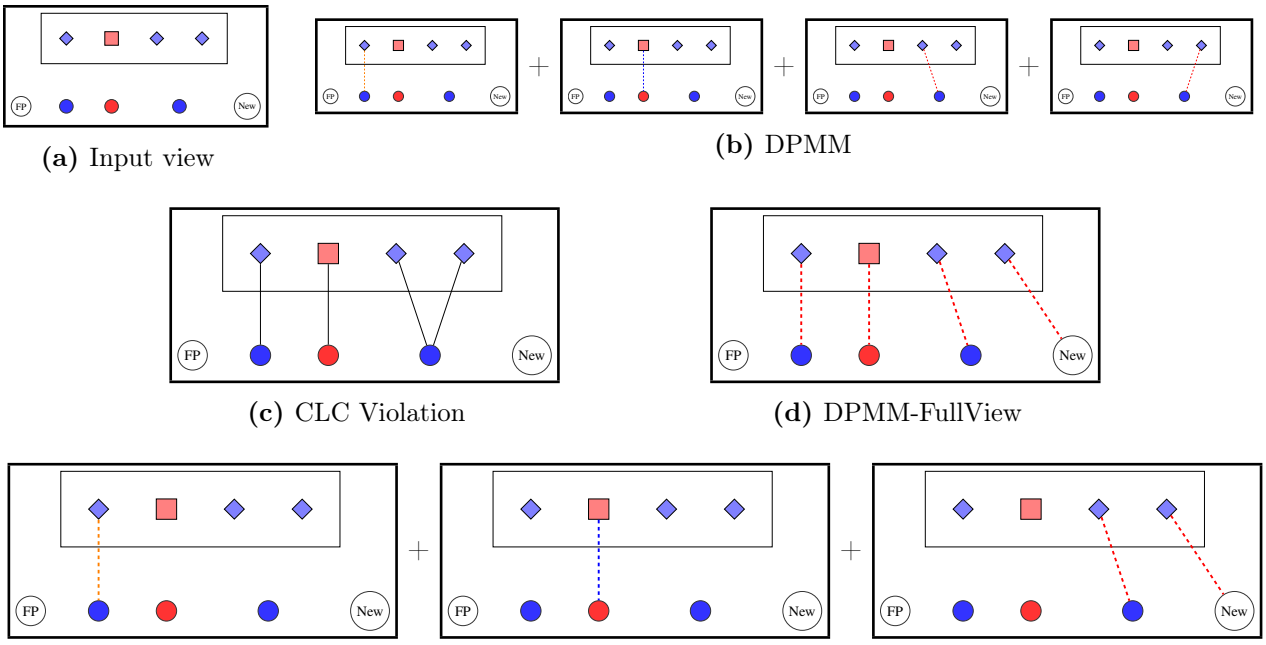


Figure 3-4: A concrete example for illustrating concepts in Section 3.5. **(a)** Each thick outer box depicts measurements in the same single view (inner box), and the clusters that each measurement can be assigned to (row below inner box). The view we consider has 4 measurements of lights' locations and colors. The 3 existing clusters within the field of view are shown as colored circles (these were determined from other views). Measurements can also be assigned to the two 'clusters' to the left and right, for false positives and new clusters respectively. The task is to assign one of the 5 clusters in the bottom row to each measurement in the inner box. **(b)** The DPMM samples cluster assignments for each measurement independently. **(c)** This causes potential violations of the cannot-link constraint (CLC), where each cluster generates at most one observation within each view. **(d)** One solution is to consider all measurement assignments in the view jointly. However, as explained in Section 3.5.1, this is inefficient. **(e)** A more efficient approximation is derived in Section 3.5.2 by jointly considering *only* measurements that are CLC-violating. Measurements that are unlikely to cause constraint violation, such as the two left ones in the example, are considered independently. This provides a trade-off between DPMM and DPMM-FullView.

porate the FN information and the CLC.

To use $\mathbb{P}_{\text{FullView}}$ as the prior over $\mathbf{z}^v$ in Equation 3.16, we must remove the assumption that we know $\{k\}^v$. The correct Bayesian approach is to integrate over the posterior distribution of the lights' locations, which are independent t-distributions, given by Equation 3.27 in the Appendix. Although this is intractable, it can be approximated by sampling the lights' locations, which is simple for t-distributions, then averaging the subsequent probabilities from Equation 3.18. In practice, I found that using the posterior mean location was sufficient, i.e., including light k if $\nu_k \in [a^v, b^v]$.

Although $\mathbb{P}_{\text{FullView}}$ combines all the desired information, the inherent difficulty is hidden in the '$\propto$' sign. The distribution first needs to be normalized before we can sample from it, which is inefficient now because the support of the distribution is the set of correspondence vectors satisfying the CLC assumption. The CLC fully couples the measurements' cluster assignments, and all assignments must be considered jointly, as depicted in Figure 3-4(d). We have essentially reverted

Input: Observations $\{\{(o_m^v, x_m^v)\}_{m=1}^{M^v}\}_{v=1}^{V}$
Fields of view $\{[a^v, b^v]\}$
Number of samples S
Output: Samples of cluster assignments $\{\{\{z_m^{v\,(s)}\}_{m=1}^{M^v}\}_{v=1}^{V}\}_{s=1}^{S}$

1: Init. $K := 0$; $z_m^{v\,(0)} := 0$ for all $m \in [M^v], v \in [V]$
2: **for** $s := 1$ **to** S; $v := 1$ **to** V **do**
3: Find cluster post./pred. distributions and sizes using most-recent samples $\{\mathbf{z}^{(s-1)}\}^{-v}$
4: Find $\{k\}^v$, the lights within field of view: include light k iff. mean location $\nu_k \in [a^v, b^v]$
5: **for each** valid correspondence vector $\mathbf{z}^v$ (from total given by Equation 3.10) **do**
6: Compute sampling 'probability' $\mathbb{P}\left(\mathbf{z}^v \,\middle|\, \{\mathbf{z}^{(s-1)}\}^{-v}, \{\{(o,x)\}\}, \{k\}^v\right)$ (unnormalized; Equations 3.8, 3.16–3.18)
7: Sample $\mathbf{z}^{v\,(s)}$ from normalized distribution
8: $K := K + n_\infty(\mathbf{z}^{v\,(s)})$
9: Remove clusters with no observations

(a) Collapsed Gibbs sampling for DPMM-FullView

Input: Observations, fields of view, num. samples
Cluster penalty parameter λ
Output: Samples of cluster assignments

1: Init. K, $\{\{z_m^{v\,(0)}\}\}$ from DP-means
2: **for** $v := 1$ **to** V **do**
3: $\mathcal{P}^v :=$ Partition induced by transitive closure of R^v, where $(i,j) \in R^v$ iff. $z_i^{v\,(0)} = z_j^{v\,(0)} \neq 0$
4: **for** $s := 1$ **to** S; $v := 1$ **to** V **do**
5: Find cluster post./pred. distributions and sizes using most-recent samples $\{\mathbf{z}^{(s-1)}\}^{-v}$
6: **for each** subset of indices $p \in \mathcal{P}^v$ **do**
7: Find assigned lights $\{k\}^v|_p$: $k \in \{k\}^v|_p$ iff. $\min_{i\in p} \|\nu_k - x_i^v\| < \min_{j\notin p} \|\nu_k - x_j^v\|$
8: Sample $\mathbf{z}^{v\,(s)}|_p$ by performing steps 5–7 of DPMM-FullView (Figure 3-5(a)), using $\{k\}^v|_p$ and $\{(o_m^v, x_m^v)\}_{m\in p}$
9: $\mathbf{z}^{v\,(s)} :=$ Concatenation of $\left\{\mathbf{z}^{v\,(s)}|_p\right\}_{p\in\mathcal{P}^v}$
10: Update clusters (DPMM-FullView steps 8–9)
11: Agglomerate elements in partitions with CLC violations (steps 2–3)

(b) Partitioning and sampling for DPMM-Factored

Figure 3-5: Modified DPMM Gibbs samplers (Figure 3-2(a)), by incorporating view-level information and constraints (DPMM-FullView), and using an efficient factored approximation (DPMM-Factored).

to the high branching factor of the MHT! In the Figure 3-4 example, $\mathbb{P}_{\text{FullView}}$ must be evaluated for 304 different values of $\mathbf{z}^v$, compared to the $4 \times 5 = 20$ required for the DPMM.

3.5.2 DPMM-Factored

A closer look at the nature of the CLC violation suggests a potential approximation to $\mathbb{P}_{\text{FullView}}$. In Figure 3-4(c), the violation is caused by *only* the two right-most measurements; the two measurements on the left are not easily confusable with the others and hence are easy to handle from a data association perspective. This suggests coupling *only* those measurements that cause CLC violations, and assume that violations involving other measurements are unlikely. Taking this a step further, we can even consider the other measurements *independently*, as in the DPMM, essentially splitting the view into three independently-considered components, as depicted in Figure 3-4(e).

More generally, suppose we can partition each view's set of measurements into 'violating' subsets, where all CLC violations are contained within a single subset, with high probability. That is, a good partition has the property that any two measurements belonging to different subsets will have low probability of being assigned to the same cluster (and hence causing an CLC violation). Let $\mathcal{P}^v$ denote such a partition on the measurement indices in view v, and let $\left\{\mathbf{z}^v|_p\right\}_{p\in\mathcal{P}^v}$ denote the restrictions of $\mathbf{z}^v$ to each subset $p \in \mathcal{P}^v$ (i.e., $\mathbf{z}^v|_p$ represents the collection of correspondence

variables $\{z_m^v\}_{m\in p}$). Then we can approximately split the CLC over the partition:

$$\mathbb{I}\Big[\mathbf{z}^v \text{ satisfies CLC}\Big] \approx \prod_{p\in\mathcal{P}^v} \mathbb{I}\Big[\mathbf{z}^v|_p \text{ satisfies CLC}\Big] \tag{3.19}$$

Returning to Figure 3-4(c), the most-CLC partition contains three subsets, where the sole non-singleton contains the two right-most CLC-violating measurements.

To make inference more efficient, we use the partition $\mathcal{P}^v$ to split $\mathbb{P}_{\text{FullView}}$. Ultimately, we want to express the right-hand side of Equation 3.18 as a *product of independent factors*, each corresponding to one subset of measurements in the partition. Because the terms are independent, sampling over the conditional distribution can be performed by sampling each factor and combining the results. Each factor is normalized over its significantly-smaller set of valid correspondence vectors, thereby improving on the bottleneck step in Gibbs sampling for DPMM-FullView.

We now consider how to factor the other two terms in $\mathbb{P}_{\text{FullView}}$ (Equation 3.18). $\mathbb{P}_{\text{DPMM}}\left(\mathbf{z}^v \,|\, \mathbf{z}^{-v}\right)$ is already a product over the conditional distributions of the correspondence variables, which is clear from Equation 3.17. By collecting terms according to the partition $\mathcal{P}^v$, we can write:

$$\mathbb{P}_{\text{DPMM}}\Big(\mathbf{z}^v \,\Big|\, \mathbf{z}^{-v}\Big) = \prod_{p\in\mathcal{P}^v} \mathbb{P}_{\text{DPMM}}\Big(\mathbf{z}^v|_p \,\Big|\, \mathbf{z}^{-vp}\Big) \tag{3.20}$$

The remaining term, $\mathbb{P}\left(\{\delta_k\}\right)$, is also a product of distributions, but over the set of lights $\{k\}^v$ that are within the field of view. Unfortunately, this cannot be immediately written as a product over the partition. We therefore make a further approximation by assigning each light to some $p \in \mathcal{P}^v$. In particular, for each light, the closest measurement (from the light's posterior mean location) was determined, and the light was assigned to the partition subset containing the measurement. Another way to view this scheme is that the partition $\mathcal{P}^v$ induces a partition over Voronoi cells in the space of location measurements (bounded by the field of view), and lights are assigned to partition elements according to the cells that their posterior mean locations are situated in.

Putting everything together, we arrive at the following *factored* approximation:

$$\mathbb{P}_{\text{Factored}}\Big(\mathbf{z}^v \,\Big|\, \mathbf{z}^{-v}, \{k\}^v\Big) \propto \prod_{p\in\mathcal{P}^v} \mathbb{P}_{\text{DPMM}}\Big(\mathbf{z}^v|_p \,\Big|\, \mathbf{z}^{-vp}\Big)\, \mathbb{P}\Big(\{\delta_k\}|_p\Big)\, \mathbb{I}\Big[\mathbf{z}^v|_p \text{ satisfies CLC}\Big] \tag{3.21}$$

where $\{\delta_k\}|_p$ denotes the restriction of $\{\delta_k\}$ to the lights assigned to subset p according to the scheme described above. This form makes clear that each factor can be normalized and sampled independently. With a good partition, the large joint computation in DPMM-FullView is broken into several smaller ones within each element of $\mathcal{P}^v$. For the concrete example in Figure 3-4, the sampling process is depicted in Figure 3-4(e), where the partition is such that only the CLC-violating measurement pair is considered jointly. This results in computing $5 + 5 + 22 = 32$ values, which is slightly greater than DPMM (20) but significantly fewer than DPMM-FullView (304).

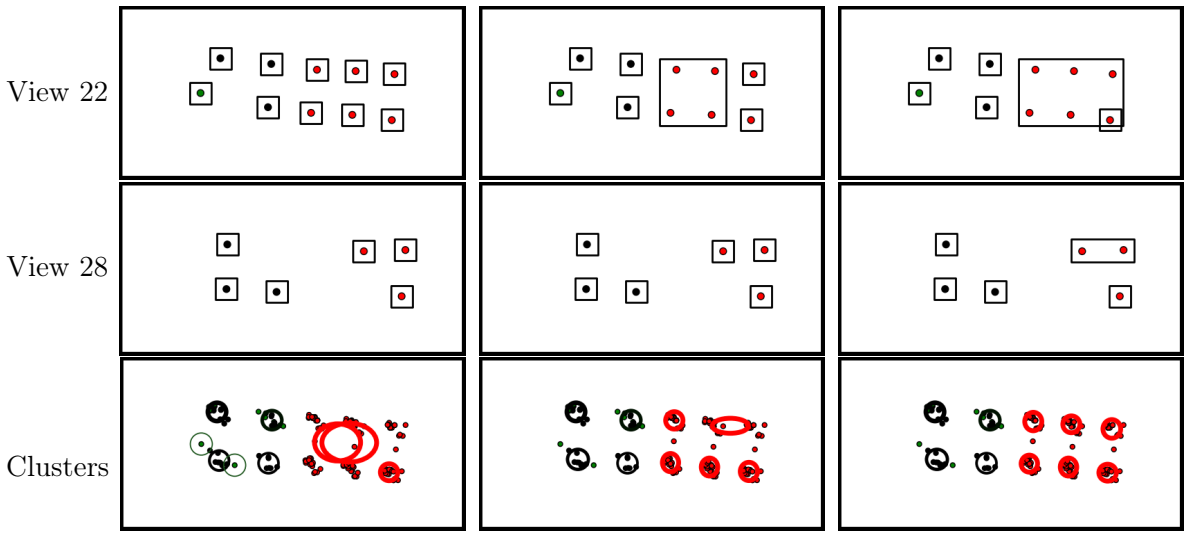
One issues remains: Where does the partition come from? This is crucial for all factored approximations: the aggressiveness of partitioning determines the trade-off between approximation

error and efficiency. On one extreme, the DPMM model is similar to a fully-factored model (but does not take into account false negatives); on the other extreme, DPMM-FullView is equivalent to a one-set partition. The example in Figure 3-4(c) once again provides an answer: 'violating' subsets can be found by examining clusters in the DPMM samples. Specifically, if measurements tend to be assigned to the same cluster across samples, then clearly they are strong violators and should be considered jointly. We therefore group measurements together if the proportion of samples in which they are assigned to the same cluster exceeds some threshold value. This proportion allows one to select an appropriate trade-off level.

For further efficiency in determining the partition, we also considered using the DP-means hard-clustering algorithm described at the end of Section 3.4. The observations were first used to quickly generate a deterministic clustering, after which the cluster assignments were examined. If two measurements within the same view were assigned to the same non-false-positive cluster, they were marked as coupled together. The partitions used by DPMM-Factored were then determined by taking the transitive closure of the coupling relation in each view. Formal details on finding this partition and determining the subsequent assigned lights can be found in Figure 3-5(b).

Returning to the real-world example from Figure 3-3, the steps taken by DPMM-Factored are partially illustrated in Figure 3-6. Initially, DP-means is used to cluster the measurements, and considers correspondences independently, which is equivalent to using a partition of singletons (boxes around each measurement in iteration 0). Like the DPMM sample shown in Figure 3-3(b), measurements in the middle from similar nearby objects are aggregated into two large clusters. The partition is updated by examining the current correspondences in each view and grouping together measurements (in the same view) that are assigned to the same cluster, i.e., violating the CLC assumption. This results in the large box for view 22 in iteration 2. Concretely, this means that during sampling, the correspondences for the four measurements are considered *jointly*, as a single product term in Equation 3.21, such that an CLC violation will not exist *for this subset of four measurements only*. This partition expansion is considered for each view, and as a result splits the large cluster into four separate ones. However, this creates a new CLC violation, so partition expansion is performed again, resulting in an even larger partition for iteration 4. This finally resolves all CLC violations and identifies objects in the scene correctly.

This example also illustrates the computational advantages of using DPMM-Factored, as opposed to DPMM-FullView and MHT. Although multiple iterations are needed to converge to a partition that resolves all CLC violations, these iterations are fast if the initial partition is aggressively fine (e.g., the all-singleton partition induced by DP-means). Our partition expansion scheme couples measurements together where necessary but no further, unlike MHT, which tends to couple more measurements even with aggressive gating, and DPMM-FullView, which couples together all measurements in a single view. For example, the four measurements on the left (black/green) tend to be sampled jointly by other approaches, but DPMM-Factored detects that they are sufficiently well-separated empirically (CLC violations are rare) and leaves them to be sampled independently. Another example is illustrated in the final iteration for view 22, where the six measurements on



(a) Iteration 0 **(b)** Iteration 2 **(c)** Iteration 4

Figure 3-6: An illustration of the DPMM-Factored method being applied on the real-world example from Figure 3-3. The top two rows show two views (of 28) taken for the scene, with different partitions of measurements (surrounding boxes) assigned by DPMM-FullView over several iterations. The final row depicts the inferred clusters/objects and their attributes (in thick ellipses) after aggregating correspondences from the 28 views. **(a)** The correspondences are initialized by DP-means, which assumes that measurements are independent, hence partitions consist of singletons only. The resulting clusters suffer from the same issues depicted in Figure 3-3, as expected. In particular, the red measurements in the middle are frequently aggregated into two large clusters, which is incorrect. **(b)** In view 22, an CLC violation is detected because the four red measurements in the middle are all previously assigned to one of the large clusters. These singleton partition elements are agglomerated to respect the assumption that CLC violations are contained within a single subset, and are sampled together to ensure their joint correspondence does not violate the CLC. This modification significantly improves the cluster estimates, with only one error remaining. The other two red measurements on the right, and the three in view 28, are not coupled at this point because they respect the CLC so far. **(c)** The two measurements in the top right now form a CLC violation. This shows that measurements that were previously not in violation could later become CLC-violating, because the clusters, and therefore the probability of being assigned to them, change between iterations. The partition is updated again to couple the CLC-violating measurements and results in the correct clustering. (The large partition element in view 22 does not contain the bottom right measurement, which is a singleton.)

the right (red) are split into a subset of five measurements, and a singleton (bottom right); other methods would consider all six together. Because the number of valid correspondence vectors to consider is combinatorial in the number of measurements (Equation 3.9), finer partitions directly imply more efficient inference (at the potential expense of accuracy). Using Equation 3.10, the total number of associations that potentially need to be considered with six measurements is $n_{\text{total}}(M^v = 6, K^v = 6) = 58576$, whereas the number for the DPMM-Factored partition is $n_{\text{total}}(M^v = 5, K^v = 5) + n_{\text{total}}(M^v = 1, K^v = 1) = 5755$, an order of magnitude less. In practice, when gating is applied for the situation shown, MHT typically evaluates 1800–2400 correspondences, whereas DPMM-Factored only considers 250-350.

3.6 Application to Object Type-and-Pose Estimation

As mentioned in Section 3.2, the colored-lights domain is representative of the semantic world-model estimation problem by considering lights as objects, and colors/locations as discrete/continuous attributes respectively. Other attributes are straightforward to incorporate as long as their predictive distributions are simple to compute. To see this, recall that in our various approaches to compute or sample from the distribution of measurement associations (Equations 3.3, 3.14, and 3.16), attribute measurements only appear in predictive likelihood terms (e.g., Equation 3.4). These predictive terms assess how well each measurement (o, x) fits with an object, whose attributes have a posterior distribution determined by all the other measurements $\{\{(o, x)\}\}_{z=k}$ currently associated with the object. Such computations are necessary for each measurement, so simplicity in calculating the predictive likelihood is crucial. However, the measured attribute values do not appear elsewhere; in particular, the correspondence priors described in Sections 3.3–3.5 do not depend on the observed values, and can be applied as described for arbitrary numbers and types of attributes.

Object attributes such as type and pose will produce predictive distributions similar in form to that for color and location in the colored-lights domain (see the Appendix for the forms). More generally, discrete attributes will have terms resembling 'color', and continuous attributes with Gaussian-like noise will have terms resembling 'location'. If attributes are independent, we can take the product of their observation models to determine the joint posterior or predictive distributions, such as the product in Equation 3.4. Dependent attributes will need to be jointly considered as a single unit. For example, for pose estimates with non-diagonal error covariances, the normal-gamma prior needs to be replaced with a normal-Wishart prior. For simplicity, I assume that the error covariance is axis-aligned and use an independent normal-gamma prior for each dimension. This is partially justified by the empirical observation that our measurements do not align significantly in any particular direction (see the small dots in Figure 3-7, depicting individual measurements).

I applied our discussed approaches to object type-and-pose estimation on tabletop scenes, illustrated in Figure 4-1. Admittedly, estimating object types and poses from separate object detections in the fashion proposed below is unlikely the most effective use of visual data, and that much information (e.g., image features, contextual cues, similarities between consecutive frames) are discarded in the process. However, I am ultimately interested in situations where only black-box attribute detectors are accessible. Object type-and-pose estimation was chosen as an exemplary problem because it is simple to understand and has immediate application.

Estimating object types and poses is similar to the colored-lights problem, where 'type' is a discrete attribute equivalent in form to 'color', and 'pose' is a 3-D version of 'location' with Gaussian-like observation noise. For our experiments, we placed a uniform prior on the 'type' attribute, with the $\mathbb{P}(\text{correct detection}) = 0.6$, $\mathbb{P}(\text{false negative}) = 0.1$, and the rest of the probability mass spread uniformly across other types. In the notation of Equation 3.1, for type i we assume that:

$$\phi_i^i = 0.6 \qquad \phi_0^i = 0.1 \qquad \phi_{j\neq i}^i = \frac{1 - \phi_i^i - \phi_0^i}{C - 1} \tag{3.22}$$

For 'pose', I assumed that objects are resting stably and upright above a surface, so only the (x, y, θ) positions of their reference points were considered. Further, as mentioned above, I assumed the observation noise is independent in each dimension, and placed a normal-gamma distribution on each, with the same hyperparameters as specified in the Appendix ($\alpha_0 = 10, \beta_0 = 9 \times 10^{-3}$). The observation likelihood of each measurement is similar in form to Equation 3.4, except with two additional terms in the product that resemble 'location' for the two extra attribute dimensions.

Detections of object type and pose came from 3-D point-cloud data obtained from a Kinect sensor mounted on a mobile robot. The object detector starts by clustering the points above a resting surface, such as a table or a shelf. It assumes that objects are separated on the surface, so that each point cloud corresponds to at most one object. For each cluster, it tries fitting point-cloud object models to the observed points, by optimizing a fitness function with respect to the object pose (x, y, θ). The z-position of the object is constrained so that the bottom of the object touches the resting surface. The optimizer tries alignments starting from several different model resting orientations based on the convex hull of the object model. The fitness function is computed as a weighted linear combination of least-squares range-image errors and point-cloud nearest-neighbor distances (approximated with a distance transform of the model point cloud). The detection pipeline is related to the system described by Glover and Popovic (2013).

For our scenarios, objects of 4 distinct types, possibly with multiple instances of the same type, were placed on a table. A robot moved around the table in a circular fashion, obtaining 20-30 views in the process; see Figures 3-9 and 3-10 for RGB images of example views (although only depth data is used to detect objects). When a view is captured, the object detection system described above is given the Kinect point cloud as input, as outputs a list of object types and (x, y, θ) poses, one for each segmented cluster in the point cloud. Figure 3-8 visualizes the detections (superimposed in red using the known shape models) for several views; common detection errors are also illustrated. I constructed 12 scenes of varying object and occlusion density to test our approaches; results for 5 representative scenarios are described in the next section.

3.6.1 Qualitative results

Qualitative results for 5 representative scenarios are shown in Figure 3-7. Images from above are for comparison convenience only; the camera's viewing height is much closer to the table height, as shown in Figures 3-9 and 3-10, so in each view only a subset of objects is detectable. I compare three approaches: multiple hypothesis tracking (**MHT** from Section 3.3; a re-implementation of the approach by Elfring et al. (2013)), generic DPMM clustering (**DPMM** from Section 3.4), and the factored approximation to DPMM-FullView (**DPMM-Factored** from Section 3.5.2). In Figure 3-7, the most likely hypothesis is shown for **MHT**, and the maximum *a posteriori* (MAP) sample (out of 100) is shown for the clustering-based approaches.

All approaches work well for scenario 1, where objects are spaced far apart. As objects of similar type are placed near each other, **DPMM** tends to combine clusters since it ignores the OMPO assumption. This is most apparent in scenario 4 (also illustrated in Figures 3-3 and 3-6),

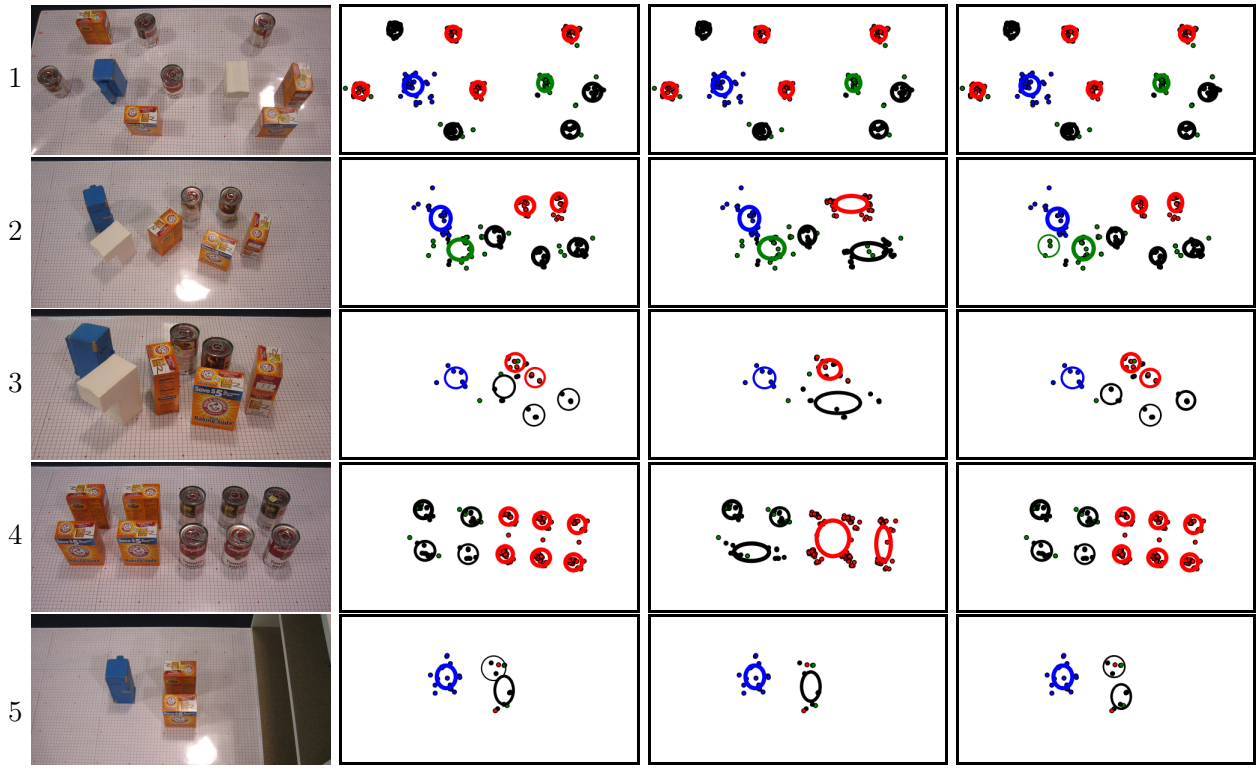
where four soup cans (red) were combined into a single large cluster. By reconsidering the OMPO assumption, **DPMM-Factored** performs significantly better and is on par qualitatively with the **MHT**, except for an extra cluster (bottom left, green) in scenario 2.

In more detail, for scenario 2, the measurements corresponding to the white L-shaped object are dispersed, causing the shown extra-cluster error to be likely. Examining more samples reveals that a significant proportion (31%) do not have the extra cluster; they just happen not to be MAP samples. This means that the estimator has significant uncertainty as to whether or not the extra object exists. Although in this case the **DPMM-Factored** MAP sample is wrong, it highlights a feature of our approach. Consider a task, e.g., grasping, that requires an accurate estimate of this object's neighborhood. Given the high uncertainty in the samples, the robot should decide to gather more observations of the region instead of operating based on the incorrect MAP sample. In contrast, the **MHT** is over 90% certain of its estimate because most other possibilities have been pruned. Although **MHT** would have been less certain as well if all hypotheses were retained during filtering, the necessary aggressive pruning tends to make **MHT** overconfident in its estimates.

Scenario 5, shown in Figure 3-9, highlights another difference between the tracking filter and batch approaches. There is significant occlusion early in the sequence, which throws off **MHT**, causing it to make incorrect associations that result in poor pose estimates. Here two closely-arranged boxes are placed near a shelf, such that from most views at most one of the two boxes can be seen. Only in the final views of the sequence can both be seen (final image in Figure 3-9). Due to the proximity of the boxes, and the fact that at most one was visible in the early views, **MHT** eventually pruned all the then-unlikely hypotheses positing that measurements came from two objects. When finally both are seen together, although a hypothesis with two orange boxes resurfaces, it is too late: the remaining association hypotheses already associate all previous measurements of the boxes to the same target, in turn giving an inaccurate location estimate. In contrast, **DPMM-Factored** re-examines previous associations (in the next sampling iteration) after the two boxes are seen together, and can correct such errors. The difference here is that the clustering-based methods repeatedly revisits all association decisions, whereas **MHT** prunes away most association hypotheses, and once having done so it cannot revisit a wrong decision.

3.6.2 Quantitative comparisons

Quantitative metrics are given in Table 3.1, averaged over the association hypotheses for **MHT**, and over 100 samples for **DPMM**, **DPMM-FullView**, and **DPMM-Factored** (after discarding burn-in). I also compare against our version of the hard-clustering algorithm **DP-means**, for several different penalty parameter (λ) settings; recall that larger λ tends to lead to more, tighter clusters (see Figure 3-2(b), line 5 to see its role as a threshold for cluster membership). Finally, I consider a baseline approach, **Raw**, that does not perform any data association. It uses the object types and poses perceived in each view directly as a separate prediction of the objects present within the visible field of view. The metrics in the table are evaluated for each view's prediction, and the **Raw** table rows show the average value over all views.

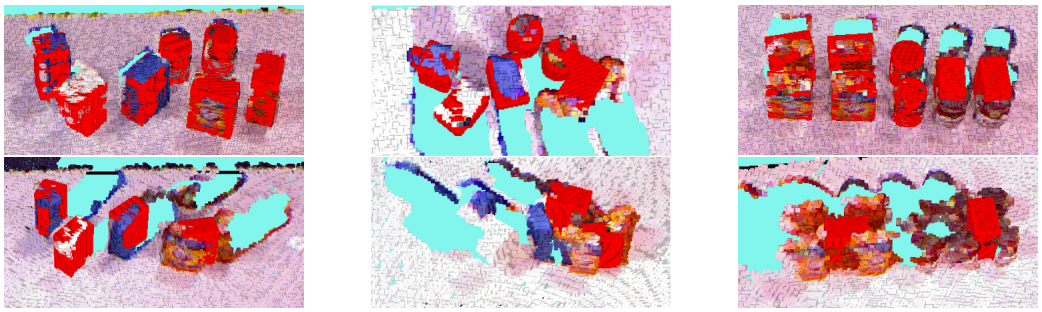


(a) Scene from above **(b)** MHT **(c)** DPMM **(d)** DPMM-Factored

Figure 3-7: Qualitative results for 3 world-model estimation approaches in 5 representative scenarios. The bird's-eye view of the scenes is for comparison convenience only; the actual viewing height is much closer to the table. The most likely hypothesis is shown for MHT, and the maximum a posteriori sample is shown for the clustering-based approaches. Each small colored dot is a semantic (object type-and-pose) detection. Each target/cluster is depicted by an ellipse, centered at the posterior mean location. Ellipse axis lengths are proportional to the standard deviation in their respective dimensions. Ellipses are color-coded by the most likely posterior object type: red = red soup can, black = orange baking-soda box, green = white L-shaped block, blue = blue rectangular cup. Line thickness is proportional to cluster size. See text in Section 3.6 for qualitative comparisons.

The need for aggregating measurements across views is exemplified by **Raw**'s tendency to miss objects or confuse their types within single views. Out of the 12 scenes attempted, scenario 3, shown in Figure 3-10, was the most challenging for **Raw**, because objects were placed close together. This caused segmentation errors frequently occur and resulted in clear errors and unidentifiable point cloud clusters. Significant occlusion also caused missed detections. As a result, using any single view's detections is unlikely to produce an accurate estimate of objects in the world. In the scenario's sequence of 21 views, although most objects are detected (sometimes incorrectly) fewer than 5 times, the combined information is sufficient for **MHT**, **DPMM-FullView**, and **DPMM-Factored** to achieve good qualitative and quantitative results.

Summarizing the results in Table 3.1, we find that **DPMM** overcomes noise in a single view by clustering across views, but still misses many objects because it ignores the OMPO assumption and agglomerates nearby similar objects. **DPMM-FullView** respects this constraint and performs



(a) Scenario 2 **(b)** Scenario 3 **(c)** Scenario 4

Figure 3-8: Examples of object detections in several views (superimposed in red using the known shape model of the detected type). The top row shows cases where the detections are relatively accurate, whereas the bottom row shows cases where most detections are missing or wrong. Missing objects are typically due to occlusion by other objects. When only a small part of an object is visible, it is often not segmented properly, which then affects the fitted poses.

Figure 3-9: Examples of views from scenario 5. In all views except the final one, only one of the two orange baking-soda boxes was detected. This causes **MHT** to incorrectly infer that all measurements came from a single object (and prune other possibilities), whereas batch approaches such as clustering can revisit previous associations and correct them using new information, such as the final view where both boxes are finally detected.

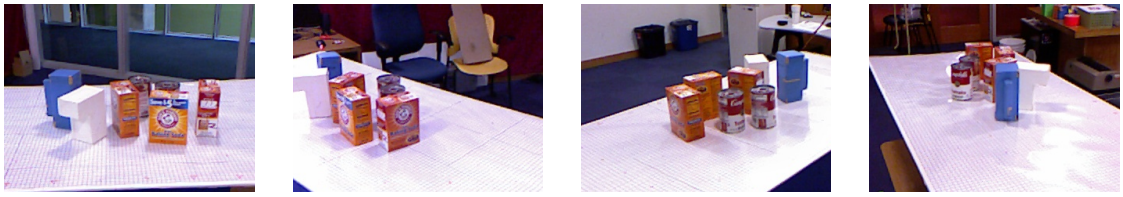


Figure 3-10: Examples of views from scenario 3. Objects were placed relatively close together, resulting in significant occlusion, causing the object detector to frequently miss detections or produce errors.

significantly better, missing few objects while maintaining accuracy in the posterior type-and-pose estimates. **DPMM-Factored** performs similarly in quality, with an increase in spurious clusters. However, this minor hit in quality comes with an order-of-magnitude computational improvement compared to **DPMM-FullView**. The clustering approaches tend to have more spurious clusters because we chose hyperparameters that encourage positing new clusters and faster exploration of the association space, but this can be corrected at the expense of convergence speed. The **MHT** achieves the overall best quantitative performance, but in most cases is only marginally better than **DPMM-Factored**, an improvement that comes at a high computational expense, and potentially introduces filtering-related overconfidence issues mentioned earlier.

Table 3.1: Average accuracy metrics and computation wall times for the five scenarios shown in Figure 3-7. **Raw** is a baseline that does not perform data association; object detections are used 'as-is'. The three 'λ' rows refer to the **DP-means** algorithm, for different settings of the penalty parameter. To evaluate predicted targets and clusters against our manually-collected ground truth, for each ground truth object, the closest cluster within a 5 cm radius is considered to be the estimate of the object. If no such cluster exists, then the object is considered missed; all predicted clusters not assigned to objects at the end of the process are considered spurious. For the two parameter accuracy metrics (most-likely type, location error), parameters were only evaluated for clusters that were matched to the ground truth (i.e., for the clusters counting towards the number of correct clusters), so values are not comparable across all methods. Computation wall times were computed on a single core of an 2.3 GHz Intel Core i7 processor, using implementations in Python. Correspondences evaluated and computation times are not provided for **Raw** since no processing in the measurements is required. The '†' symbol for **MHT** and **DPMM-FullView** in scenarios 1 and 4 indicate that external guidance was necessary, in the form of manually splitting the views (each into 2-3 parts, consisting of 3-6 measurements each). Without such help, both algorithms take over an hour to complete for scenarios 1 and 4; scenario 2 is an example of a case of moderate complexity, where no guidance was provided.

Metric →	Num. correct objects (TPs)					Num. missed objects (FNs)					Num. spurious clusters (FPs)					F_1 score $= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$				
Scenario	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Raw	8.0	3.3	1.6	5.3	1.0	2.0	3.7	5.4	4.7	2.0	0.8	1.3	0.3	0.1	0.7	0.85	0.53	0.32	0.64	0.74
$\lambda = -1$	4.0	2.0	2.0	1.0	0.0	6.0	5.0	5.0	9.0	3.0	1.0	1.0	**0.0**	2.0	1.0	0.53	0.40	0.44	0.15	0.00
$\lambda = -2.5$	8.0	2.0	5.0	4.0	3.0	2.0	5.0	2.0	6.0	**0.0**	2.0	4.0	**0.0**	1.0	**0.0**	0.80	0.31	0.83	0.53	**1.00**
$\lambda = -4$	**10.0**	**7.0**	6.0	**6.0**	**3.0**	**0.0**	**0.0**	**1.0**	4.0	**0.0**	4.0	5.0	2.0	3.0	1.0	0.83	0.74	0.80	0.63	0.86
MHT	**10.0**	**7.0**	**6.0**	**10.0**	2.4	**0.0**	**0.0**	**1.0**	**0.0**	0.6	**0.0**	**0.0**	**0.0**	**0.0**	0.6	**1.00**	**1.00**	**0.92**	**1.00**	0.81
DPMM	8.0	2.1	2.1	4.0	2.7	2.0	4.9	4.9	6.0	0.3	1.0	2.7	**0.0**	1.0	**0.0**	0.84	0.36	0.46	0.53	0.94
FullView	**10.0**	**7.0**	**6.0**	**10.0**	**3.0**	**0.0**	**0.0**	**1.0**	**0.0**	**0.0**	0.1	1.5	0.2	**0.0**	**0.0**	**1.00**	0.91	0.91	**1.00**	**1.00**
Factored	**10.0**	**7.0**	**6.0**	**10.0**	2.9	**0.0**	**0.0**	**1.0**	**0.0**	0.1	0.8	3.1	0.3	**0.0**	**0.0**	0.96	0.82	0.90	**1.00**	0.97

Metric →	Most-likely type is correct (%)					Location estimate error (cm)					Num. correspondences evaluated ($\times 10^3$)					Computation wall time (s)				
Scenario	1	2	3	4	5	1	2	3	4	5	1†	2	3	4†	5	1†	2	3	4†	5
Raw	98	93	67	85	56	2.5	2.7	1.9	2.2	2.1	N/A					N/A				
$\lambda = -1$	100	100	50	100	-	2.0	2.5	3.8	1.1	-	4.48	2.50	0.21	1.70	0.03	24.3	7.4	1.0	8.6	0.6
$\lambda = -2.5$	100	100	80	100	100	2.2	2.4	2.4	1.0	2.2	9.09	2.65	0.76	2.16	0.25	19.4	6.2	0.5	6.7	0.6
$\lambda = -4$	100	100	100	100	100	2.1	2.6	1.7	1.3	2.4	12.7	5.58	0.86	3.80	0.21	12.8	5.3	0.1	4.0	0.1
MHT	100	100	83	100	100	2.1	2.8	1.8	1.3	2.6	5.72†	195	18.3	105†	0.25	16.9†	593	41.6	211†	0.5
DPMM	100	100	56	100	95	2.2	2.4	3.8	1.0	2.7	69.9	23.5	3.37	22.0	2.88	144	39.8	3.6	37.3	4.2
FullView	100	100	90	100	100	2.0	2.6	1.6	1.4	2.2	44.9†	582	19.3	167†	2.62	171†	1346	33.7	278†	5.3
Factored	100	100	88	100	96	2.1	2.6	1.6	1.3	2.4	8.91	6.84	1.48	19.0	0.98	127	64.9	9.9	90.6	4.5

3.7 Discussion

I have presented several clustering-based data association approaches for estimating semantic world models. I use Dirichlet process mixture models (DPMM) as our underlying framework. However, DPMMs perform poorly in their generic form because they ignore crucial view-level information and constraints. Two improvements were therefore developed by incorporating the cannot-link constraint (CLC) exactly and approximately respectively. In preliminary experiments based on tabletop object type-and-pose estimation, the latter approach (**DPMM-Factored**) achieved performance comparable to a tracking-based approach (**MHT**) using a fraction of the computation time.

If only a single posterior association is needed (instead of a distribution), the hard-clustering algorithm **DP-means** performs surprisingly well, and is much faster than all the other methods. However, performance depends heavily on setting this parameter appropriately. To some extent, this could be alleviated by starting with an overly-conservative (large) value of λ, with few clusters and many CLC violations, and then gradually decreasing λ until most violations are resolved. This will still lead to an abundance of spurious clusters, as seen in the results. Using view-level information to merge *clusters* (instead of sampling *cluster assignments*) may prove beneficial.

Besides small-variance asymptotics methods such as **DP-means**, there exist many other fast inference algorithms for DPMMs. Part of the motivation for applying DPMMs to data association problems is that, once a computational 'reduction' is achieved between the problems, then inference tools in DPMMs and clustering can be translated into algorithms for data association. Interesting avenues to explore include split-merge MCMC procedures (Jain and Neal, 2004; Dahl, 2005), search algorithms (Daumé III, 2007; Wang and Dunson, 2011), variational inference algorithms (Blei and Jordan, 2006; Pacheco and Sudderth, 2012), and others (e.g., Raykov et al., 2014).

As discussed in the introduction, semantic world models are useful in many object-centric tasks, involving a diverse set of attributes. The model in this chapter technically supports attributes beyond object type and pose, but actual applications will be needed to support this claim. To be truly applicable, world models must also cope with objects moving over extended periods of time. Since the presented sampling procedure for inference iterates through all views, it is at present impractical to apply it to the entirety of the robot's observation history. Instead, a hybrid approach combining the benefits of filtering and batch data association is desirable. Extending the clustering-based data association framework to handle temporal dynamics is the subject of the next chapter.

3.A Appendix: Derivation of posterior and predictive distributions for cluster parameters

In this appendix, I verify the claim from Section 3.2 that finding the posterior and predictive distributions on color and location for a single light is straightforward, given that we know which observations were generated by that light. Let $\{(o, x)\}$ denote the set of light color-location detections that correspond to a light with unknown parameters (c, l). Color and location measurements are assumed to be independent given (c, l) and will be considered separately. I assume a known discrete prior distribution $\pi \in \Delta^{(C-1)}$ on colors, reflecting their relative prevalence. Using the color noise model (Equation 3.1), the posterior and predictive distributions on c are:

$$\mathbb{P}\left(c \,|\, \{o\}\right) \propto \mathbb{P}\left(\{o\} \,|\, c\right)\ \mathbb{P}\left(c\right) \propto \left[\prod_{i \in \{o\}} \phi_i^c\right]\ \pi_c \tag{3.23}$$

$$\mathbb{P}\left(o' \,\middle|\, \{o\}\right) = \sum_{c=1}^{C} \mathbb{P}\left(o' \middle| c\right)\ \mathbb{P}\left(c | \{o\}\right) = \sum_{c=1}^{C} \phi_{o'}^c\ \mathbb{P}\left(c \,|\, \{o\}\right) \tag{3.24}$$

We can use this to find the light's probability of detection:

$$p_{\mathrm{D}} \triangleq 1 - \mathbb{P}\left(o' = 0 \,\middle|\, \{o\}\right) = 1 - \sum_{c=1}^{C} \phi_0^c\ \mathbb{P}\left(c \,|\, \{o\}\right) \tag{3.25}$$

Unlike the constant false positive rate p_{FP}, the detection (and false negative) rate is dependent on the light's color posterior.

For location measurements, we emphasize that both the mean l *and* precision $\tau = \frac{1}{\sigma^2}$ of the Gaussian noise model is unknown. Modeling the variance as unknown allows us to attain a better representation of the location estimate's empirical uncertainty, and not naïvely assume that repeated measurements give a known fixed reduction in uncertainty each time. I use a standard conjugate prior, the distribution NormalGamma$(l, \tau; \lambda, \nu, \alpha, \beta)$. The typical interpretation of normal-gamma hyperparameters is that the mean is estimated from λ observations with mean ν, and the precision from 2α observations with mean ν and variance $\frac{\beta}{\alpha}$. It is well known (e.g., Bernardo and Smith (1994)) that after observing n observations with sample mean $\hat{\mu}$ and sample variance $\hat{s}^2$, the posterior is a normal-gamma distribution with hyperparameters:

$$\begin{aligned} \lambda' &= \lambda + n & \nu' &= \frac{\lambda}{\lambda + n}\nu + \frac{n}{\lambda + n}\hat{\mu} \\ \alpha' &= \alpha + \frac{n}{2} & \beta' &= \beta + \frac{1}{2}\left(n\hat{s}^2 + \frac{\lambda n}{\lambda + n}\left(\hat{\mu} - \nu\right)^2\right) \end{aligned} \tag{3.26}$$

Often we are only interested in the posterior distribution of the mean; the marginal distribution

on μ is a three-parameter (degrees of freedom, mean, scale) non-standardized t-distribution:

$$\mathbb{P}\left(l \,|\, \{x\} \;;\; \lambda, \nu, \alpha, \beta\right) = \text{StudentT}\left(l \,;\, 2\alpha', \nu', \sqrt{\frac{\beta'}{\lambda'\alpha'}}\right) \tag{3.27}$$

where the normal-gamma hyperparameters have been updated using $\{x\}$ according to Equation 3.26. Prior to any observations, the hyperparameters are set to $\lambda_0 = 0, \nu_0 = 0$ (representing a noninformative prior over location) and α_0, β_0 chosen such that $\frac{\beta_0}{\alpha_0}$ is equal to a prior value of the variance, using α_0 to toggle the prior strength. For location, we use $\alpha_0 = 10$ and $\frac{\beta_0}{\alpha_0} = 9 \times 10^{-4}$, representing a weak prior where the location standard deviation is expected to be around 3cm.

The upshot of using a conjugate prior for location measurements is that the marginal likelihood of location observations has a closed-form expression. The posterior predictive distribution for the next location observation x' is obtained by integrating out the latent parameters l, τ:

$$\begin{aligned}
\mathbb{P}\left(x' \,\middle|\, \{x\} \;;\; \lambda, \nu, \alpha, \beta\right) &= \int_{(l,\tau)} \mathbb{P}\left(x \,|\, l, \tau\right) \mathbb{P}\left(l, \tau \,|\, \{x\} \;;\; \nu, \lambda, \alpha, \beta\right) \\
&= \int_{(l,\tau)} \mathcal{N}\left(x \,;\, l, \tau^{-1}\right) \text{ NormalGamma}\left(l, \tau \,;\, \nu^-, \lambda^-, \alpha^-, \beta^-\right) \\
&= \frac{1}{\sqrt{2\pi}} \frac{\beta^{-\alpha^-}}{\beta^{+\alpha^+}} \frac{\sqrt{\lambda^-}}{\sqrt{\lambda^+}} \frac{\Gamma(\alpha^+)}{\Gamma(\alpha^-)}
\end{aligned} \tag{3.28}$$

where hyperparameters with "$-$" superscripts are updated according to Equation 3.26 using the empirical statistics of $\{x\}$ only (excluding x'), and ones with "$+$" superscripts are likewise updated but including x'. The ratio in Equation 3.28 assesses the fit of x' with the existing observations $\{x\}$ associated with the light.

Chapter 4

Object-based World Modeling in Semi-Static Environments with Dependent Dirichlet Process Mixtures

By definition, objects have the potential to be manipulated. This means that the static world model from the previous chapter will almost certainly be incorrect over time. When we considered the data association problem that is at the core of semantic world modeling, methods for multiple target tracking such as the MHT were capable of solving the problem, although we found superior clustering-based solutions. Targets in conventional applications such as aircraft tracking typically move every time step, so conventional data association methods are capable of tracking the state of the world over time. The goal of this chapter is to extend the clustering-based solutions, which appeared promising in the static case, to also work in environments that change over time.

In multiple-target tracking problems, each target's state (typically location) changes between observations. However, if we consider applications such as tracking objects in a household, the dynamics are typically different: most objects tend to stay in the same state when they are not being actively used. In this chapter, we therefore study the world modeling problem in *semi-static environments*, where time is divided into known epochs, and within each epoch the world is stationary. It seems intuitive that data association should be easier within static periods, since there is no uncertainty arising from stochastic dynamics.

An alternative approach to data association is to perform inference over the entire time-series of observations and to think of it as a problem of clustering: we wish to group together similar detections over time, under the assumption that they were been generated by the same individual. Bayesian nonparametric models, such as the Dirichlet process mixture model (DPMM), can be used to model domains in which the number of individuals is unknown *a priori*; in the previous chapter, I found that a state-estimation technique based on DPMM clustering was effective for determining the number and type of objects in a static domain, given a sequence of images with partial views of the scene and significant occlusions within them.

In this chapter, I apply the clustering approach to the much more difficult case of a dynamic domain in which the attributes of objects may change over time, new objects may appear, and old objects may permanently disappear. The DPMM is not an appropriate model for this problem, but an extension, the *dependent Dirichlet process mixture model* (DDPMM), which models dependencies between a collection of clusters, can be used effectively. In particular, I use a construction proposed by Lin et al. (2010) for a class of DDPs that can be represented as a Markov chain over DPs; an overview of this model can be found in Section 2.2.3. In our case of semi-static world modeling, objects in each static epoch are modeled as clusters in a DPMM (as in the previous chapter), and clusters between epochs are related by Markovian transitions, thus forming a DDPMM.

In the remainder, I will formalize the world modeling problem, review the DDP construction and apply it to our problem, and derive a novel approximate maximum *a posteriori* (MAP) inference algorithm for the model. I show that this model yields computational advantages for tracking in semi-static environments, both in simulation and on real-world data.

4.1 Problem Definition

In world modeling, we seek the state of the world, consisting an unknown finite number K^t of objects, which changes over time. Object k at epoch t has attribute values θ^{kt}. We sometimes decompose θ^{kt} into $\left(a^k, x^{kt}\right)$, where a is a vector of fixed attributes, and x is a vector of attributes that may change between epochs. The top row in Figure 4-1 illustrates the world state over three epochs for a simple domain.

Our system obtains noisy, partial views of the world. Each view v produces a set of observations $O^{tv} = \left\{o_i^{tv}\right\}$, where $o_i^{tv} = \left(b_i^{tv}, y_i^{tv}\right)$, corresponding to the fixed attributes a and dynamic attributes x^t of some (possibly non-existent) object[1]. Each view is also associated with a field of view V^{tv}. The collection of views in a single epoch may fail to cover the entire world. The partial views and noisy observations are illustrated in the middle and bottom rows of Figure 4-1.

The world modeling problem can now be defined: Given observations $O = \left\{o_i^{tv}\right\}_{(t,v,i)}$ and fields of view $\left\{V^{tv}\right\}_{(t,v)}$, determine the state of objects over time $\Theta = \left\{\theta^{kt}\right\}_{(k,t)}$. The state includes not only objects' attribute values, but also the total number of objects that existed at each epoch, and implicitly when objects were added and removed (if at all).

There is no definitive information in the observations that will allow us to know which particular observations correspond with which underlying objects in the world, or even how many objects were in existence at any time step. For example, in the views of $t = 1$ shown in Figure 4-1, the square detected in the left-most view may correspond to either (or neither) square in the center view. Also, despite there being only four objects in the world, there were five observations because of overlapping visible regions.

The critical piece of information that is missing is the *association* z_i^{tv} of an observation o_i^{tv} to an underlying object k. With this information, we can perform statistical aggregation of the

[1] Superscripts in variables will generally refer to the 'context', such as object index k and time index t. Subscripts refer to the index in a list, such as $o_i^t = i$'th observation at time t.

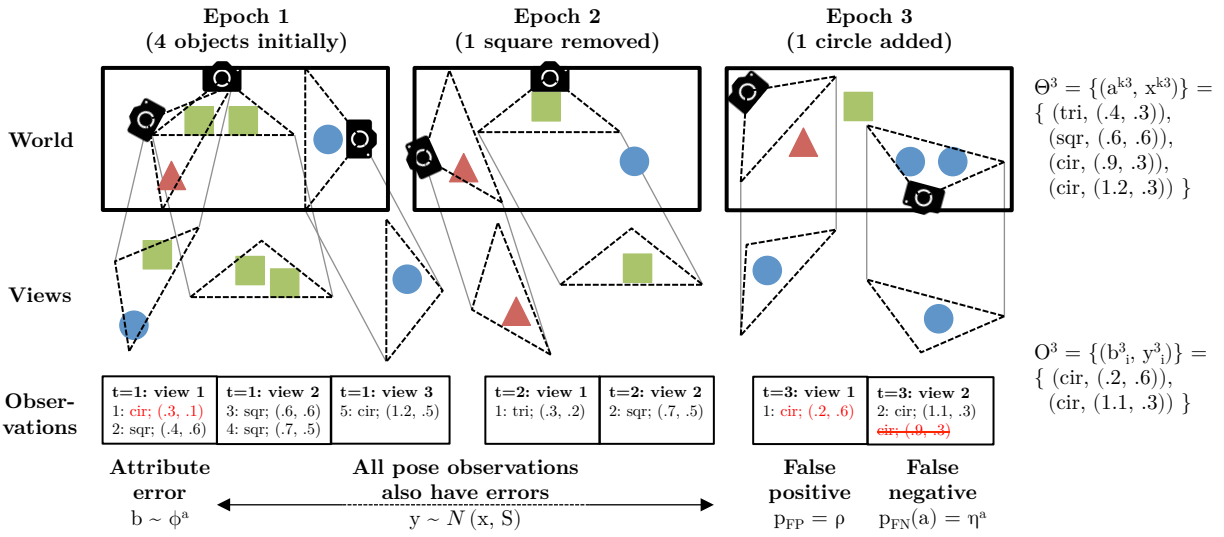


Figure 4-1: An illustration of the world modeling problem. An unknown number of objects exist in the world (top row), and change in pose and number over time (world at each epoch enclosed in box). At each epoch, limited views of the world are captured, as depicted by the triangular viewcones. Within these viewcones, objects and their attributes are detected using black-box perception modules (e.g., off-the-shelf object detectors). In this example, the attributes are shape type (discrete) and 2-D location. The observations are noisy, as depicted by the perturbed versions of viewcones in the middle row. Uncertainty exists both in the attribute values and the existence of objects, as detections may include false positives and negatives (e.g., $t = 3$). The actual attribute detection values obtained from the views are shown in the bottom row ("Observations"); this is the format of input data. Given these noisy measurements as input, the goal is to determine which objects were in existence at each epoch, their attribute values (e.g., Θ^3 in top right), and their progression over time.

observations assigned to the same object to recover its state. I will model the associations $Z = \left\{z_i^{tv}\right\}_{(t,v,i)}$ as latent variables in a Bayesian inference process.

4.1.1 Observation noise model

The observation model describes how likely an observation $o = (b, y)$ was generated from some given object state $\theta = (a, x)$ (if any), given by the probability $f(o\,;\,\theta)$. For a single object, let θ_c and θ_d be the true continuous and discrete attribute values respectively, and likewise o_c and o_d for a single observation of the object. We typically consider observation noise models of the following form:

$$f(o\,;\,\theta) = \phi^{\theta_d}(o_d)\,\mathcal{N}(o_c\,;\,\theta_c, S) \tag{4.1}$$

Here ϕ represents a discrete confusion matrix, where $\phi^{\theta_d}(o_d)$ is the probability of observing o_d given the true object has discrete attributes θ_d. The continuous-valued observation o_c is the true value θ_c corrupted with zero-mean Gaussian noise, with fixed sensing covariance S. The noise on o_c and o_d are assumed to be independent for simplicity.

Besides errors in attribute values, Figure 4-1 also illustrates cases of false positives and false negatives. A false positive occurs when the observation did not originate from any true object. I assume that this occurs at a fixed rate ρ. When this occurs, o_d has noise distribution ϕ^0, and o_c is uniformly distributed over the field of view V. A false negative occurs when an object is within the sensor's field of view but failed to be detected. I assume that an object within the field of view V will be undetected with an attribute-dependent probability $\eta(\theta)$.

4.1.2 Additional assumption: Cannot-link constraint (CLC)

As in the previous chapter, there is an additional common domain assumption in target-tracking problems: within a single view, each visible object can generate at most one detection (Bar-Shalom and Fortmann, 1988). This implies that within a view, each observation must be assigned to a different hypothesized underlying object. We previously referred to this as the "cannot-link constraint" (CLC). The constraint is powerful because it can reduce ambiguities when there are similar nearby objects. However, similar to the DPMM modifications proposed in Section 3.5, we will need to modify the DDPMM model and inference algorithms to handle the CLC.

4.2 A Clustering-Based Approach

I now specify a prior on how likely an assignment to a cluster is, and how clusters change over time. Since the number of clusters are unknown, I chose to use Bayesian nonparametric mixture models (see Section 2.2 for an overview), which allow for an indefinite and unbounded number of mixture components (although the number of instantiated components is limited by the data size).

We now restate one result of the DDP construction; see Section 2.2.3 and Lin (2012) for details. The construction results in the following prior on parameter θ^t (to be assigned to a new observation), given past parameters $\Theta^{<t}$ and parameters Θ^t corresponding to clusters that have already been instantiated at the current epoch (compare with Equation 2.37):

$$\theta^0 \mid \Theta^0 \propto \alpha H\left(\theta^0\right) + \sum_k N^{k0}\, \delta_{\theta^{k0}}\left(\theta^0\right) \tag{4.2}$$

$$\theta^t \mid \Theta^{\leq t} \propto \alpha H\left(\theta^t\right) + \sum_{k:N^{kt}>0} N^{k,\leq t}\, \delta_{\theta^{kt}}\left(\theta^t\right) + \sum_{k:N^{kt}=0} q(\theta^{k,t-1})\, N^{k,<t}\, T\left(\theta^t\,;\, \theta^{k,t-1}\right)$$

At the initial time step, clusters are formed as in a standard DPMM with concentration parameter α and base distribution H. For later time steps, the prior distribution on θ is defined recursively. The first two terms are similar to the base case, for new clusters and already-instantiated clusters (in the current epoch) respectively. The third term corresponds to previously-existing clusters that may be removed with probability $(1 - q(\theta^{k,t-1}))$, and, if it survives, is moved with transition probability $T\left(\cdot\,;\, \theta^{k,t-1}\right)$. $N^{k,\leq t}$ is the number of points that have been assigned to cluster k, for all time steps up to time t. Note that if $q \equiv 1$ and $T\left(\cdot\,;\,\theta\right) = \delta_\theta$, then the model is static, and Equation 4.2 is equivalent to the predictive distribution in the DP.

4.2.1 Inference by forward sampling

As mentioned in the problem definition, our focus will be on determining latent assignments $Z = \left\{z_i^t\right\}$ of observations $O = \left\{o_i^t\right\}$ to clusters with parameters $\Theta = \left\{\theta^{kt}\right\}$. In the generic DDP, views do not exist yet; those will be introduced in Section 4.3. One way to explore the distribution of assignments is to sample repeatedly from the assignment's conditional distribution, given all other assignments $Z_{\setminus ti} \triangleq Z \setminus \left\{z_i^t\right\}$:

$$\begin{aligned}\mathbb{P}\left(z_i^t = k \,\middle|\, o_i^t, \Theta, Z_{\setminus ti}\right) &= \int \mathbb{P}\left(z_i^t = k, \theta \,\middle|\, o_i^t, \Theta, Z_{\setminus ti}\right)\, \mathrm{d}\theta \\ &\propto \int \mathbb{P}\left(o_i^t \,\middle|\, \theta\right) \mathbb{P}\left(\theta = \theta^{kt} \,\middle|\, \Theta, Z_{\setminus ti}\right)\, \mathrm{d}\theta \end{aligned} \tag{4.3}$$

The first term in the integrand is given by the observation noise model (Equation 4.1), and the second term is given by the DDP prior (Equation 4.2). If θ^{kt} already exists, then $\mathbb{P}\left(\theta \,\middle|\, \Theta, Z_{\setminus ti}\right) = \mathbb{I}\left[\theta = \theta^{kt}\right]$, and the integrand only has support for $\theta = \theta^{kt}$. Otherwise, we have to consider all possible settings of θ^{kt}, which has a prior distribution given by Equation 4.2. The expression in Equation 4.3 above can be decomposed into three cases, corresponding to terms in Equation 4.2:

$$\mathbb{P}\left(z_i^t = k \,\middle|\, o_i^t, \Theta^{\leq t}, Z_{\setminus ti}^{\leq t}\right) \propto \begin{cases} N_{\setminus ti}^{k,\leq t} & f\left(o_i^t\,;\,\theta^{kt}\right)\ , \\ & k \text{ existing, instantiated at } t \\ \tilde{q}(\theta^{k\tau})\ N_{\setminus ti}^{k,<t} & \int f\left(o_i^t\,;\,\theta\right)\ \tilde{T}\left(\theta\,;\,\theta^{k\tau}\right)\, \mathrm{d}\theta\ , \\ & k \text{ existing, not instantiated at } t \\ \alpha & \int f\left(o_i^t\,;\,\theta\right)\ H\left(\theta\right)\, \mathrm{d}\theta\ , \\ & k \text{ new} \end{cases} \tag{4.4}$$

In the DDPMM, clusters move around the parameter space during their lifetimes, and, depending on our chosen viewpoints, may not generate observations at some epochs. When cluster k has at least one time-t observation assigned to it, it becomes *instantiated* at time t. Any subsequent observations at time t that are assigned to cluster k must then share the same parameter θ^{kt}; this corresponds to the first case. The second case is for clusters not yet instantiated at time t, and we must infer θ^{kt} from the last known parameter for cluster k, at time $\tau < t$. If $t - \tau > 1$, we use generalized survival and transition expressions for our application (see Equation 2.37, and the discussion before that, for further explanation):

$$\begin{aligned} \tilde{q}(\theta^{k\tau}) &\triangleq \left[q(\theta^{k\tau})\right]^{t-\tau} \\ \tilde{T}\left(\theta^{kt}\,;\,\theta^{k\tau}\right) &= \mathbb{I}\left[a^{kt} = a^{k\tau}\right]\ \mathcal{N}\left(x^{kt}\,;\,x^{k\tau}, (t-\tau)R(a^k)\right) \end{aligned} \tag{4.5}$$

The third case is for new clusters that are added at time t. The first and third cases above correspond to the existing-cluster and new-cluster cases in the DP respectively (see Equation 2.33).

In general, since the cluster parameters Θ are also unknown, inference schemes need to alternate between sampling the cluster assignments (given parameters) as above, and sampling the parameters given the cluster assignments. The conditional distribution of each cluster's parameters $\left\{\theta^{kt}\right\}$ (for each cluster k, a sequence of parameters) can be found using Bayes' rule:

$$\mathbb{P}\left(\left\{\theta^{kt}\right\} \middle| O, Z\right) = \mathbb{P}\left(\left\{\theta^{kt}\right\} \middle| O|_{z=k}, Z\right) \propto \left[\prod_{z_i^t=k} \mathbb{P}\left(o_i^t \middle| \theta^{kt}\right)\right] \mathbb{P}\left(\left\{\theta^{kt}\right\}\right) \tag{4.6}$$

Depending on the choice of parameter priors and observation functions, the resulting conditional distributions can potentially be complicated to represent and difficult to sample from. With additional assumptions that will be presented next, we can find the parameter posterior distribution efficiently and avoid sampling the parameter entirely by 'collapsing' it.

4.2.2 Application of DDPs to world modeling

We now apply the DDP mixture model (DDPMM) to our semi-static world modeling problem. For concreteness and simplicity, we consider an instance of the world modeling problem where the fixed attribute a is the discrete object type (from a finite list of known types), and the dynamic attribute x is the continuous pose in $\mathbb{R}^d$ (either 3-D location or 6-D pose). Despite these restrictions, our model and derivations below can be immediately applied to problems with any fixed attributes, and with any dynamic continuous attributes with linear-Gaussian dynamics. Arbitrary dynamic attributes can be represented in our model, but inference will likely be more challenging because in general we will not obtain closed-form expressions.

For our instance of the DDPMM, we assume:

- Time steps in the DDP correspond to epochs in world modeling. This implies that each epoch is modeled as a static DPMM, similar to the case in the previous chapter.

- The survival rate only depends on the fixed attribute, i.e., $q(\theta) = q(a)$. (For us, that means the likelihood of object removal is dependent on the object type but not its pose.)

- Likewise, the detection probability only depends on the fixed attribute, i.e., $\eta(\theta) = \eta(a)$.

- The dynamic attribute (pose) follows a random walk with zero-mean Gaussian noise that depends on a (e.g., a mug likely travels farther per epoch than a table):

$$x^{t+1} = x^t + w, \text{ where } w \sim \mathcal{N}\left(0, R(a)\right) \tag{4.7}$$

  This implies that the full transition distribution (of both object type and pose) is:

$$T\left(\theta^{t+1}; \theta^t\right) = \mathbb{I}\left[a^{t+1} = a^t\right] \mathcal{N}\left(x^{t+1}; x^t, R(a)\right) \tag{4.8}$$

- At each epoch, the DP base distribution has the following form:

$$H(\theta) \triangleq \pi(a)\, \mathcal{N}\left(x\,;\, \mu^0, \Sigma^0\right) \tag{4.9}$$

  Here a (discrete) prior π over the object type, and a normal distribution over the object pose. The initial covariance Σ^0 is large, in order to give reasonable likelihood of an object being introduced at any location. In fact, we will set $\Sigma^0 = \infty I$ and $\mu^0 = \mathbf{0}$, representing a noninformative prior over the location. Details can be found in Appendix 4.A.

The above choices for the dynamics and base distribution implies that the parameter posterior and predictive distributions have closed-form expressions. The posterior distribution of the dynamic attribute is a mixture of Gaussians, with a component for each possible value of the fixed attribute a (since the process noise $R(a)$ may be different), weighted by the posterior probability of a. In practice, we track the pose using only the dynamics of the most-likely object type $\hat{a}^k$. Thus, in our application, each cluster will maintain a discrete posterior distribution $\varphi(a)$ for the object type, and a single Kalman filter (Kalman, 1960) / Rauch-Tung-Striebel (RTS) smoother (Rauch et al., 1965) for the object pose distribution. The latter is represented as a sequence of means and covariances $\left\{\mu^t, \Sigma^t\right\}_{t=\xi}^{\zeta}$ over the cluster's lifetime $t \in [\xi, \zeta]$, with the interpretation that $x^t \sim \mathcal{N}\left(\mu^t, \Sigma^t\right)$. More details and derivation of the parameter distributions can be found in Appendix 4.A.

As mentioned previously, because we have compact representations of the parameter posterior distributions, we can analytically integrate them out instead of sampling them. We first modify the forward sampling equation (Equation 4.4) to reflect this 'collapsing' operation. Since we can no longer condition on the parameters themselves, we instead need to condition on the other observations $O_{\backslash ti}$ and their current cluster assignments $Z_{\backslash ti}$, and use posterior *predictive* likelihoods of the form $\mathbb{P}\left(o_i^t \,\middle|\, O_{\backslash ti}^k\right)$ to evaluate the current observation o_i^t:

$$\begin{aligned}
\mathbb{P}\left(z_i^t = k \,\middle|\, o_i^t, O_{\backslash ti}^{\leq t}, Z_{\backslash ti}^{\leq t}\right) &\propto \mathbb{P}\left(o_i^t \,\middle|\, z_i^t = k, O_{\backslash ti}^{\leq t}, Z_{\backslash ti}^{\leq t}\right)\ \mathbb{P}\left(z_i^t = k \,\middle|\, O_{\backslash ti}^{\leq t}, Z_{\backslash ti}^{\leq t}\right) \\
&\propto \int \left[\mathbb{P}\left(o_i^t \,\middle|\, \theta^{kt}\right)\ \mathbb{P}\left(\theta^{kt} \,\middle|\, O_{\backslash ti}^{k,\leq t}\right)\right]\ \mathbb{P}\left(z_i^t = k \,\middle|\, Z_{\backslash ti}^{\leq t}\right)\ \mathrm{d}\theta^{kt} \\
&\propto \begin{cases}
N_{\backslash ti}^{k,\leq t} & \int \mathbb{P}\left(o_i^t \,\middle|\, \theta^{kt}\right)\ \mathbb{P}\left(\theta^{kt} \,\middle|\, O_{\backslash ti}^{k,\leq t}\right)\ \mathrm{d}\theta^{kt}\ , \\
& k \text{ existing, instantiated at } t \\
N_{\backslash ti}^{k,<t} & \int \mathbb{P}\left(o_i^t \,\middle|\, \theta^{kt}\right)\ \left[\int \tilde{q}(a^{k\tau})\, \tilde{T}\left(\theta^{kt}\,;\, \theta^{k\tau}\right)\ \mathbb{P}\left(\theta^{k\tau} \,\middle|\, O_{\backslash ti}^{k,<t}\right)\ \mathrm{d}\theta^{k\tau}\right]\ \mathrm{d}\theta^{kt}\ , \\
& k \text{ existing, not instantiated at } t \\
\alpha & \int \mathbb{P}\left(o_i^t \,\middle|\, \theta^{kt}\right)\ H\left(\theta^{kt}\right)\ \mathrm{d}\theta^{kt}\ , \\
& k \text{ new}
\end{cases}
\end{aligned} \tag{4.10}$$

We can now substitute the expressions for $\mathbb{P}\left(o_i^t \,\middle|\, \theta^{kt}\right)$, $\tilde{T}$, and H, where properties of the normal distribution will help us evaluate the integrals. The derivations in Appendix 4.A give the following

expressions, as well as details for finding the posterior hyperparameters φ, μ^{kt}, and Σ^{kt} (recall $\theta^{kt} = (a^k, x^{kt}), o_i^t = (b_i^t, y_i^t)$):

$$\mathbb{P}\left(z_i^t = k \,\middle|\, o_i^t, O_{\setminus ti}^{\leq t}, Z_{\setminus ti}^{\leq t}\right) \propto \begin{cases} N_{\setminus ti}^{k,\leq t} & \left[\sum_{a^k} \phi^{a^k}(b_i^t)\, \varphi(a^k)\right] \mathcal{N}\left(y_i^t ; \mu^{kt}, \Sigma^{kt} + S\right) , \\ & k \text{ existing, instantiated at } t \\ \tilde{q}(\hat{a}^k)\, N_{\setminus ti}^{k,<t} & \left[\sum_{a^k} \phi^{a^k}(b_i^t)\, \varphi(a^k)\right] \mathcal{N}\left(y_i^t ; \mu^{k\tau}, \Sigma^{k\tau} + (t-\tau)R(\hat{a}^k) + S\right) , \\ & k \text{ existing, not instantiated at } t \\ \alpha & \left[\sum_{a^k} \phi^{a^k}(b_i^t)\, \pi(a^k)\right] \text{Unif(vol(world))} , \\ & k \text{ new} \end{cases} \tag{4.11}$$

In the second case, for tractability in filtering, we have assumed that a cluster's dynamics behaves according to its most-likely type $\hat{a}^k$; otherwise, the posterior is a mixture of Gaussians (over all possible transition densities). Also, the third case contains an approximation to avoid evaluating an improper probability density; see Appendix 4.A for details.

4.3 Incorporating World Modeling Constraints

So far, we have only applied a generic DDPMM to our observations, but have ignored the cannot-link constraint, as well as false positives and negatives. I now present modifications to the Gibbs sampler to handle these constraints; the modifications are similar to those from the static case in the previous chapter (see Section 3.5).

The cannot-link constraint, described in Section 4.1.2, couples together cluster assignments for observations within the same view, since we must ensure that no two observations can be assigned to the same existing cluster. For each view, all cluster assignments must be considered together as a joint correspondence vector, and the probability of choosing one such correspondence is proportional to the product of the individual cluster assignment probabilities given in Equation 4.11. Invalid correspondence vectors that violate the cannot-link constraint are assigned zero probability and hence are not considered; the remaining conditional probabilities are normalized. This can be interpreted as performing *blocked* Gibbs sampling, where blocks are determined by the joint constraints:

$$\mathbb{P}\left(\mathbf{z}^{tv} \,\middle|\, \mathbf{o}^{tv}, O_{\setminus tv}, Z_{\setminus tv}\right) \propto \left[\prod_i \mathbb{P}\left(z_i^{tv} \,\middle|\, o_i^{tv}, O_{\setminus tv}, Z_{\setminus tv}\right)\right] \mathbb{I}\left[\mathbf{z}^{tv} \text{ satisfies CLC}\right] \tag{4.12}$$

The *correspondence vector* $\mathbf{z}^{tv}$ is again the concatenation of the individual z_i^{tv} assignment variables, for all observation indices i made in view v at epoch t; the interpretation of $\mathbf{o}^{tv}$ is similar. The individual terms in the product are given by Equation 4.11 (the value of z_i^{tv} determines which case

applies), except now all observations within the same view are excluded (since their assignments are being sampled together) – $O_{\backslash tv}$ instead of $O_{\backslash ti}$, and likewise for assignments $Z_{\backslash tv}$ and counts $N_{\backslash tv}$.

For false positives, I essentially treat it as a special 'cluster' that has no underlying parameter Instead, I assume that if an observation is generated from a false positive, it is generated from some spurious parameter drawn from the base distribution H, so the likelihood term is the same as that for drawing a new cluster. Like the other cases, we also multiply the likelihood by the number of points already assigned to the cluster, i.e., the number of false positives except for those in the current view. If there are currently no other false positives, then we multiply by the concentration parameter α instead to ensure that it is always feasible to assign observations to the false positive 'cluster'. Also, to incorporate the assumption that false positives are generated with a fixed rate ρ, we attach a Bernoulli probability to each case in the Gibbs sampler. The false positive conditional probability is multiplied by ρ, and all other cases are multiplied by $(1-\rho)$. In summary, the conditional probability of an observation being a false positive ($z=0$) is:

$$\mathbb{P}\left(z_i^{tv}=0 \,\middle|\, o_i^{tv}, O_{\backslash tv}, Z_{\backslash tv}\right) \propto \left[\sum_{a^k} \phi^{a^k}(b_i^{tv})\ \pi(a^k)\right]\ \text{Unif}(\text{vol}(\text{world})) \times \begin{cases} \rho N^0_{\backslash tv}\,, & N^0_{\backslash tv} > 0 \\ \rho\alpha\,, & N^0_{\backslash tv} = 0 \end{cases} \tag{4.13}$$

The normalizer depends on the other cases in Equation 4.11 (with additional $(1-\rho)$ factors).

Finally, for false negatives, recall that an object that is within the field of view fails to be detected with type-dependent probability $\eta(a^k)$. Let δ_k^{tv} be 1 if cluster k is detected in view v at epoch t, and 0 otherwise. For a cluster k that is alive at epoch t ($\xi^k \le t \le \zeta^k$) with parameter θ^{kt}, the probability of detection is therefore:

$$\mathbb{P}\left(\delta_k^{tv}=1\right) = \left[1-\eta(a^k)\right]\ \mathbb{P}\left(\theta^{kt} \in V^{tv}\right) = \left[1-\sum_{a^k}\eta(a^k)\ \varphi(a^k)\right]\ \tilde{\Phi}\left(x^{kt} \in V^{tv}\,;\, \mu^{kt}, \Sigma^{kt}\right) \tag{4.14}$$

The $\tilde{\Phi}$ function denotes the CDF of the multivariate normal distribution, with mean μ^{kt} and covariance Σ^{kt}. For a particular view V^{tv}, we only evaluate the above detection probability on clusters that are currently alive at epoch t. For each such cluster, there is a corresponding δ_k^{tv} detection indicator variable, whose value is determined during sampling by the candidate joint correspondence vector $\mathbf{z}^{tv}$: if some element of $\mathbf{z}^{tv}$ is assigned to cluster index k, then $\delta_k^{tv}=1$; otherwise, $\delta_k^{tv}=0$. The detection probability for the correspondence vector is:

$$\mathbb{P}_{\text{D}}\left(\mathbf{z}^{tv} \,\middle|\, O_{\backslash tv}, Z_{\backslash tv}\right) = \prod_{k:\ \xi^k \le t \le \zeta^k} \left[\mathbb{P}\left(\delta_k^{tv}=1\right)\right]^{\delta_k^{tv}} \left[1-\mathbb{P}\left(\delta_k^{tv}=1\right)\right]^{1-\delta_k^{tv}} \tag{4.15}$$

Putting everything together, we arrive at a constrained blocked collapsed Gibbs sampling inference algorithm. The algorithm takes the observations $O=\left\{o_i^{tv}\right\}$ and visible regions $\left\{V^{tv}\right\}$ as input. As output, the algorithm produces samples from the posterior distribution over correspondence vectors $\left\{\mathbf{z}^{tv}\right\}$, from which we can compute the posterior parameter distributions $a^k \sim \varphi$ and $x^{kt} \sim \mathcal{N}\left(\mu^{kt}, \Sigma^{kt}\right)$. The sampling algorithm repeatedly iterates over epochs t and views v, each

time sampling a new correspondence vector $\mathbf{z}^{tv}$ from its constrained conditional distribution:

$$\begin{aligned}
\mathbb{P}_{\text{View}}\left(\mathbf{z}^{tv} \mid \mathbf{o}^{tv}, O_{\setminus tv}, Z_{\setminus tv}\right) \propto & \left[\prod_{i:\, z_i^{tv}\neq 0} (1-\rho)\ \mathbb{P}\left(z_i^{tv} \mid o_i^{tv}, O_{\setminus tv}, Z_{\setminus tv}\right)\right] \\
& \times \left[\prod_{i:\, z_i^{tv}=0} \rho\ \left[\sum_{a^k} \phi^{a^k}(b_i^{tv})\ \pi(a^k)\right]\ \frac{1}{\text{vol(world)}} \times \begin{cases} N^0_{\setminus tv}\,, & N^0_{\setminus tv} > 0 \\ \alpha\,, & N^0_{\setminus tv} = 0 \end{cases}\right] \\
& \times \left[\prod_{k:\, \xi^k \leq t \leq \zeta^k} \left[\mathbb{P}\left(\delta_k^{tv}=1\right)\right]^{\delta_k^{tv}} \left[1-\mathbb{P}\left(\delta_k^{tv}=1\right)\right]^{1-\delta_k^{tv}}\right] \\
& \times \mathbb{I}\left[\mathbf{z}^{tv} \text{ satisfies CLC}\right] \qquad (4.16)
\end{aligned}$$

The probability terms in the first and third lines can be found in Equations 4.11 and 4.14 respectively.

As in the static case, after incorporating the world modeling constraints, inference becomes inefficient because we now have to compute conditional probabilities for (and sample from) the joint space of correspondence vectors, which in general is exponential in the number of observations in a view. Using the same insights and ideas as before, however, we can adaptively factor the correspondence vector by initially decoupling all assignment variables, then coupling only those that violate the cannot-link constraint; see Section 3.5.2 for details.

4.4 Approximate Maximum *a Posteriori* (MAP) inference

We have now presented the entire Gibbs sampling algorithm for DDPMM-based world modeling. However, sampling-based inference can be slow, especially because of the cannot-link constraint that couples together many latent variables, even if adaptive factoring is used. Although we are interested in maintaining an estimate of our uncertainty in the world, frequently just having the most-likely (maximum *a posteriori* – MAP) world state suffices. In general, even the MAP world model is hard to find, because it can be formulated as a multidimensional assignment problem, and it is known to be NP-complete for $T \geq 3$ epochs (Karp, 1972). Nevertheless, since it is a fundamental combinatorial problem, many approximate solutions have been proposed.

In the static case, we adapted a hard-clustering algorithm, DP-means, and empirically found that it returned good clustering assignments for some hyperparameter settings (see Algorithm 3-2(b)). A similar analysis via small-variance asymptotics was performed recently for DDPs, where the mixture components were Gaussian distributions with isotropic noise, resulting in the Dynamic Means algorithm (Campbell et al., 2013). However, there is no simple and principled way to incorporate the additional information from Section 4.3. Additionally, even without such modifications, the Dynamic Means algorithm requires three free hyperparameters to be specified, which may be significantly harder to tune than the one in DP-means. Instead, we will use a much older idea that does not involve asymptotics, can incorporate all the world-modeling information and constraints, and produces an local optimization algorithm that is similar in spirit to Dynamic Means.

4.4.1 Iterated conditional modes (ICM)

The *iterated conditional modes* (ICM) algorithm performs coordinate ascent on each variable's conditional distribution, and is guaranteed to converge to a local maximum (Besag, 1986). In particular, instead of iteratively sampling correspondence vectors from their conditional distributions in Gibbs sampling, we find the most-likely one, update parameters based on it, and repeat for each view. Since we are still dealing with the joint space of assignments for all observations in a given view, finding the maximizer still potentially requires searching through a combinatorial space. Fortunately, finding the most-likely correspondence within a single view can be formulated as a maximum weighted assignment problem, for which polynomial-time (cubic) exact algorithms such as the Hungarian algorithm exist (Kuhn, 1955; Munkres, 1957).

Suppose, for view v at epoch t, there are M observations $\{o_1, \ldots, o_M\}$ and K existing clusters (possibly not alive/instantiated). Then we wish to match each o_i to an existing cluster, a new cluster, or a false positive. Any unmatched existing cluster must also be assigned the probability of missed detection. We can solve this as an assignment problem with the following payoff matrix:

	Obs (M)	FN ($M+K$)
Clusters (K)	$\log \mathbb{P}(z_i = k) + \log(1-\rho)$ $+ \mathbb{I}\left[\xi^k \le t \le \zeta^k\right] \log \mathbb{P}(\delta_k = 1)$	$\mathbb{I}\left[\xi^k \le t \le \zeta^k\right] \log \mathbb{P}(\delta_k = 0)$
New (M)	$\log \mathbb{P}(z_i = \text{new}) + \log(1-\rho)$	0
FP (M)	$\log \mathbb{P}(z_i = 0\ (\text{FP})) + \log \rho$	0

The payoff matrix has $2M + K$ entries (indicated in parentheses), to allow for the case that all observations are assigned to new clusters, and likewise that all are spurious. Any extra New/FP nodes are assigned to extra FN nodes, with zero payoff. The payoffs in the first column are: for an existing cluster, given by cases 1 and 2 in Equation 4.11, depending on whether or not the cluster has been instantiated yet; for a new cluster, given by case 3 in Equation 4.11; and for a false positive, given by Equation 4.13. Note that log probabilities are used to decompose the view's joint correspondence probability into a sum of individual terms. By construction, the cannot-link constraint is satisfied. Besides returning the optimal assignment, algorithms such as Murty's k-best (Murty, 1968) can return multiple good assignments with slightly higher computational complexity.

4.4.2 A two-stage inference scheme

Although the ICM algorithm presented can find good clusters at a single epoch very quickly, we will see in experiments that it does not converge to good cluster trajectories. The issue is that ICM moves are local, in that it considers one view at a time. Suppose we have identified correctly all objects in epoch 1 using ICM. When we consider the first view in epoch 2, there may be significant changes present, and using observations from the first view only, ICM must decide whether or not to assign the new observations to existing clusters (by reviving them from the previous epoch). Since the uncertainty in the object states immediately after a transition is high, basing the cluster connectivity decisions on a single view is unreliable.

Input: Observations $O = \left\{o_i^{tv}\right\}$, Visible regions $\left\{V^{tv}\right\}$, Number of samples N
Output: Samples of cluster assignments $\left\{\mathbf{z}^{tv}\right\}$

1: Init. all entries to -1 (FP) in $Z^{(0)} = \left\{\mathbf{z}^{tv}\right\}^{(0)}$
2: **repeat**
3: **for** $t := 1$ **to** T; $v := 1$ **to** V^t **do**
4: Solve ICM weighted assignment problem for most-likely $\mathbf{z}^{tv}$, given $Z^t_{\setminus v}$
5: **until** convergence
6: Construct new dataset $C = \left\{c_i^t\right\}$ with a single data point for each non-FP cluster found by ICM (at the same epoch)
7: Sample tracks L by performing MCMCDA on C (Algorithm 2-1)
8: Convert track samples to cluster assignments

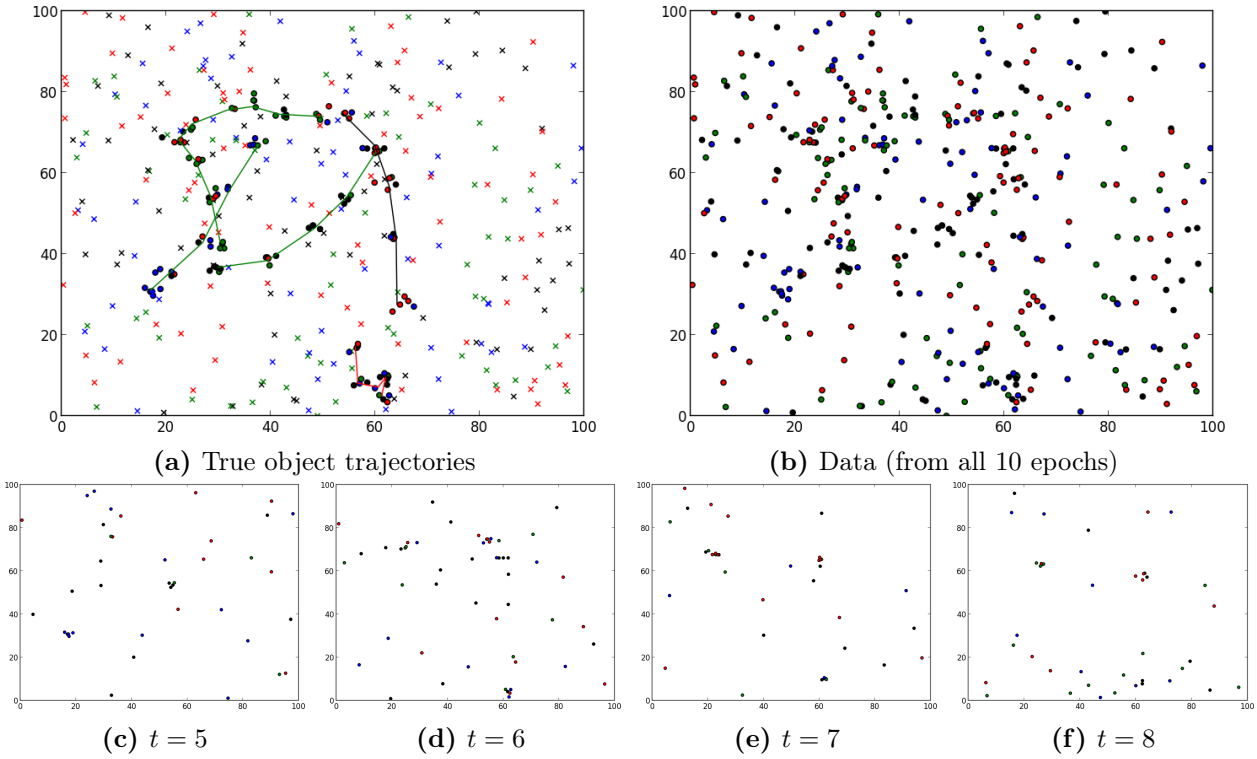
Figure 4-2: Two-stage inference algorithm for DDPMM, using ICM and MCMCDA.

This suggests a two-level inference scheme. Since ICM can reliably find good clusters within single epochs, we first apply ICM to each epoch's data *independently*, treating them as unrelated static worlds. Next, we attempt to connect clusters between different epochs. This is essentially another tracking problem, although the likelihood function is somewhat different (depends on many underlying data points), and is much reduced in size. Since the problem is significantly smaller, traditional tracking methods such as MHT can be applied to this cluster-level tracking problem.

We present one such scheme in Algorithm 4-2, using MCMCDA (Algorithm 2-1; Oh et al. (2009)) to solve the cluster-level problem. We choose a batch-mode sampling algorithm such as MCMCDA because it can return samples from the posterior distribution, and has an attractive anytime property – we can terminate at any point and still return a list of valid samples. For inferring the MAP configuration, the best sample can be returned instead. Since we are sampling from the true posterior distribution (assuming that the per-epoch clusters are identified correctly), in the limit of infinite samples, the true MAP configuration will be found almost surely.

To apply MCMCDA, we need to evaluate the likelihood of a complete configuration Z, encompassing all epochs and views (line 4 in Algorithm 2-1). To do so, we first find the posterior parameter distributions for the clusters/objects (as given by Z) using Appendix 4.A, then combine the observation likelihoods (Equation 4.32), as well as the false positive and false negative priors:

$$\begin{aligned}
\mathbb{P}\left(O \mid Z\right)\mathbb{P}\left(Z\right) &= \prod_t \prod_v \mathbb{P}\left(\mathbf{o}^{tv} \,\middle|\, \mathbf{z}^{tv}\right)\ \mathbb{P}_{\mathrm{FP}}\left(\mathbf{z}^{tv}\right)\ \mathbb{P}_{\mathrm{FN}}\left(\mathbf{z}^{tv}\right) \\
&= \prod_t \prod_v \left\{ \left[\prod_i \int \mathbb{P}\left(o_i^t \,\middle|\, \theta^{kt}, z_i^t = k\right)\ \mathbb{P}\left(\theta^{kt}\right)\ \mathrm{d}\theta^{kt}\right] \right. \\
&\qquad \times \mathrm{Bin}\left(N^{tv}_{z=0} \,\middle|\, N^{tv}, \rho\right) \\
&\qquad \left. \times \left[\prod_{k:\ \xi^k \le t \le \zeta^k} \left[\mathbb{P}\left(\delta_k^{tv} = 1\right)\right]^{\delta_k^{tv}} \left[1 - \mathbb{P}\left(\delta_k^{tv} = 1\right)\right]^{1-\delta_k^{tv}}\right] \right\} \qquad (4.17)
\end{aligned}$$



(a) True object trajectories **(b)** Data (from all 10 epochs)

(c) $t = 5$ **(d)** $t = 6$ **(e)** $t = 7$ **(f)** $t = 8$

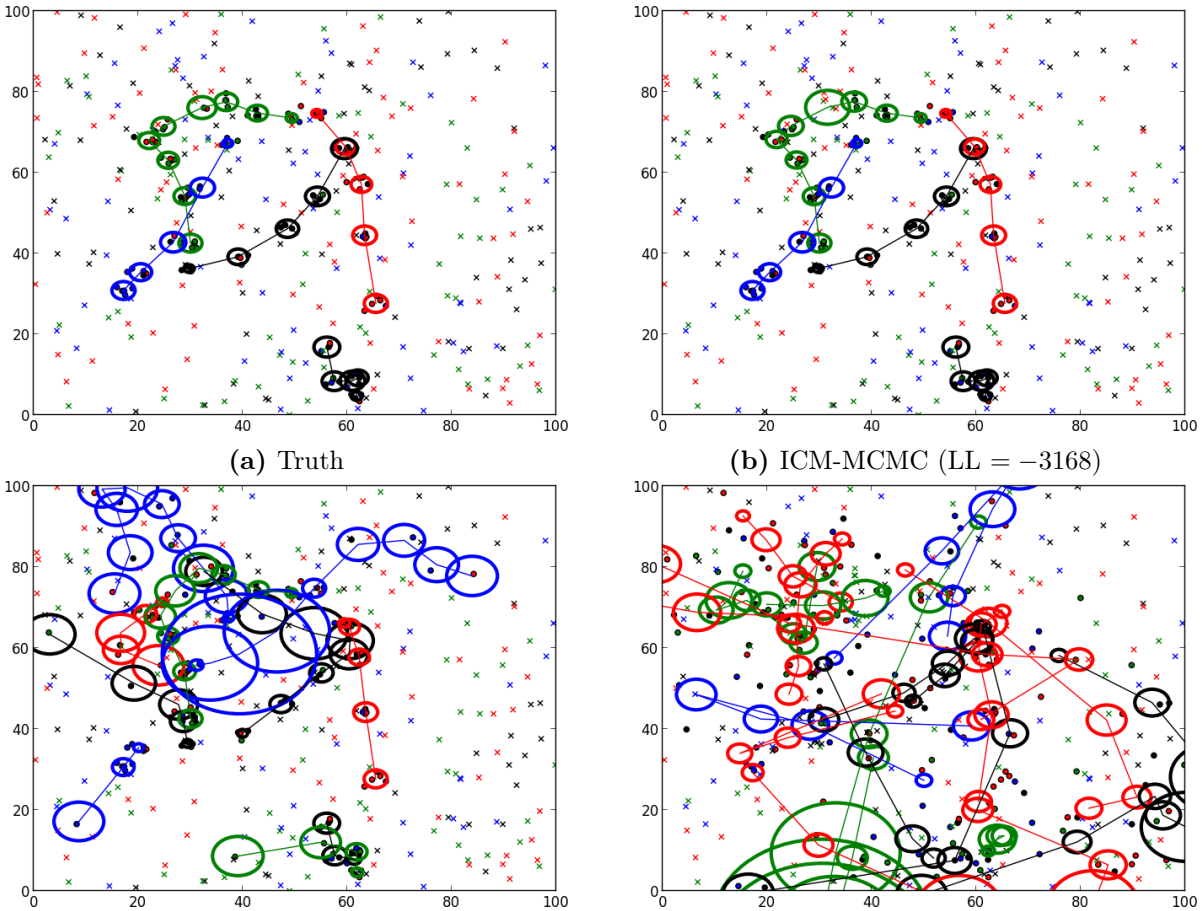
Figure 4-3: Data and object states in a simulated domain. The top left shows the true object (x, y) locations and their trajectories over time, color-coded by their associated object type. Observations are shown as filled dots (corresponding to true positives) and crosses (false positives). The top right shows the data from all 10 epochs (5 views per epoch) that is given as input, without any information about the underlying object states and associations. Some form of clustering over views and time is visible. A more realistic view of the data is shown in the bottom row, for a sequence of 4 epochs.

4.5 Experiments

Approximate MAP inference for world modeling via ICM, MCMCDA, and the two-stage algorithm ICM-MCMC were tested on a simulated domain, and also on a sequence of real robot vision data constructed from the static scenes in the previous chapter. To perform MAP inference on MCMCDA and ICM-MCMC, the most-likely sample (as scored by Equation 4.17) was chosen, from 10^5 samples in MCMCDA, and 10^4 in the second stage of ICM-MCMC. In both experiments, ICM-MCMC significantly outperforms the other two methods, and even ICM performs better than MCMCDA.

4.5.1 Simulation

Objects in our simulated domain had one of four fixed object types, a time-evolving location $(x, y) \in [0, 100] \times [0, 100]$, and a time-evolving velocity vector. Observations were made in 10 epochs of this domain, with 5 views per epoch (visible region is the entire domain). In total, 5 objects existed, each for some contiguous sub-interval of the elapsed time. Within each view, the number of false positives was generated from Poi(5), and the probability of a missed detection was 0.1. The correct object type was observed with probability 0.6, with equal likelihood (0.1) of being confused with



(a) Truth

(b) ICM-MCMC (LL = −3168)

(c) ICM (LL = −3329)

(d) MCMCDA (LL = −3365)

Figure 4-4: The clusters found for the simulated domain are shown in thick ellipses, centered at the location mean, color-coded by the most-likely object type inferred (across the entire trajectory, since it is a static attribute). The ellipses depict a level set of the posterior location distribution (uncertainty given by Gaussian covariance matrix). The posterior clusters derived from the true association is shown in the top left; the one found by ICM-MCMC is essentially identical (with a minor difference in the green track). In contrast, the posterior clusters found by ICM and the most-likely sample from MCMC (of 10^5), shown in the bottom row, are qualitatively much different, and have significantly lower log-likelihood (LL) values.

the other 3 object types. Locations were observed with isotropic Gaussian noise, standard deviation 1.0. The object's velocity vector was maintained from the previous time step, with added Gaussian noise, standard deviation 5.0. Between epochs, the probability of survival was 0.9. The observed data (i.e., the algorithm input) and the true object states are shown in Figure 4-3.

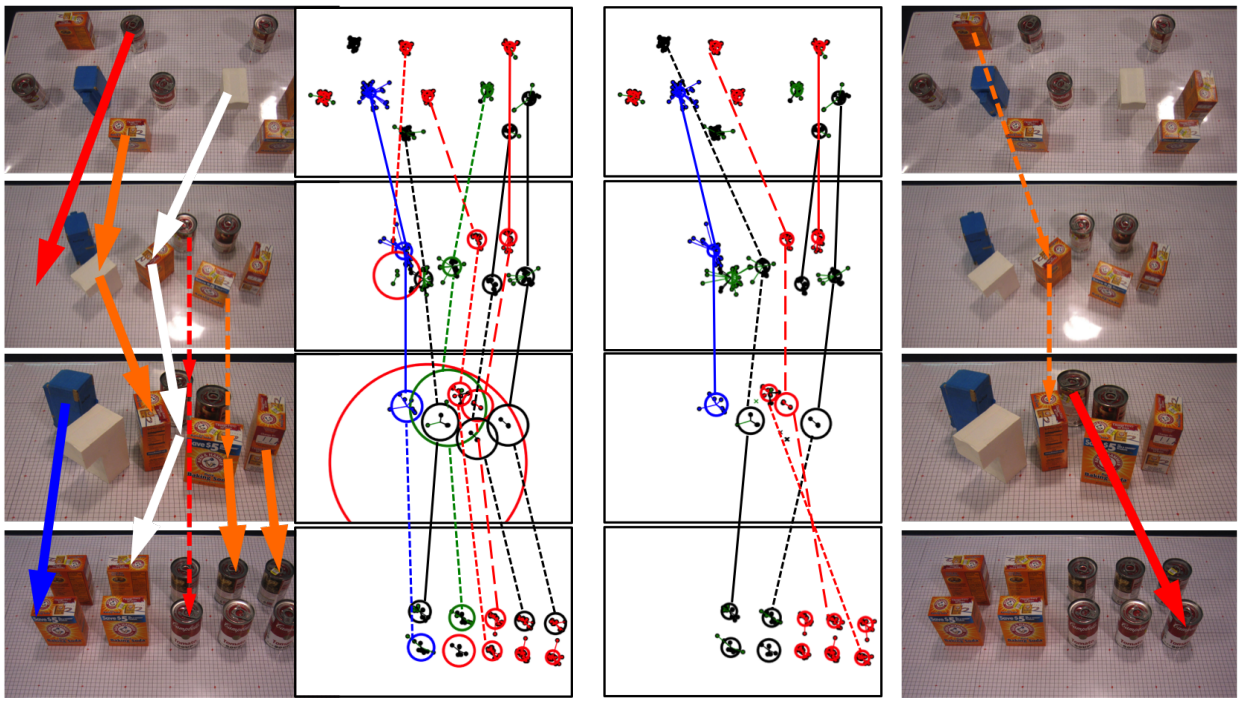
The resulting MAP clusters found by ICM, MCMCDA, and ICM-MCMC are shown in Figure 4-4, along with their log-likelihood values (higher / less negative is better). ICM-MCMC clearly outperforms the other methods, and finds essentially the same clusters as given by the true association. The clusters found generally have tight covariance values, unlike those in ICM and MCMCDA. These two methods, especially MCMCDA, tend to find many more clusters than are truly present.

4.5.2 Using robot data from static scenes

We also applied the same algorithms to the static robot vision data that were used in the previous chapter to evaluate DPMM methods. To convert static scenes into dynamic scenes, we choose static scenes that were reasonably similar, and simply concatenated their data together, as if each scene corresponded to a different epoch. One such example is shown in Figure 4-5.

Objects in different scenes were all placed on the same tabletop of dimensions 1.2m $\times$ 0.6m; all data were placed in the table's frame of reference. Four object types were present, and typically each scene had 5–10 objects. Unlike the previous simulation, we do not assume objects have velocities; between epochs, we assume that the location changes with isotropic Gaussian noise, standard deviation 0.1. Since changes were significant between epochs, we assumed a relatively low 0.5 probability of survival. Object locations are sensed with Gaussian noise, standard deviation 0.03; the object type noise model and probability of detection is the same as before. The probability of false positives is much lower for this domain; we assumed the number of false positives had a Poi(0.1) distribution.

Figure 4-5 shows the MAP associations found by ICM and ICM-MCMC, with lines connecting cluster states over epochs. Annotations were also added (in the form of three different line styles) to facilitate comparison between the ICM and ICM-MCMC results; see figure caption for details. ICM tends to suggest many more transitions than ICM-MCMC, many of which are actually implausible.



(a) ICM transitions not present in ICM-MCMC **(b)** Most-likely ICM configuration (LL = −968) **(c)** Most-likely ICM-MCMC conf (LL = −931) **(d)** ICM-MCMC transitions not present in ICM

Figure 4-5: Approximate MAP cluster (object) trajectories found using ICM and ICM-MCMC on the robot vision data collection in the previous chapter. The concatenated sequence of scenes (epochs) is shown from top to bottom. The inferred clusters and tracks are shown in the middle two columns. Lines connecting cluster pairs between epochs are color-coded by the inferred object type (fixed across epochs), and are marked by one of three line styles used to compare results from the two algorithms. A solid line means the same pair was connected by both algorithms; a dashed line means a similar pair (in likelihood) was connected; a dotted line means the pair was not connected by the other algorithm. To make the differences clearer, the top-down reference views have been annotated with arrows, for pairs of objects that were only connected by one algorithm (dotted lines in the middle two). The left column shows pairs that were connected by ICM but not ICM-MCMC; the right column shows the opposite. Solid arrows depict transitions that are unlikely, whereas dashed arrows depict plausible transitions. ICM tends to suggest many more transitions than ICM-MCMC, many of which are actually implausible.

4.6 Discussion

I have presented an extension of the clustering-based data association method in the previous chapter to semi-static environments, by applying the dependent Dirichlet process mixture model (DDPMM). Similar to the case for DPMMs, multiple modifications on the generic DDPMM were necessary to adhere to assumptions in the world modeling problem. Additionally, because of the extra temporal dimension, inference is even more challenging. A fast approximate MAP inference algorithm, iterated conditional modes (ICM), was therefore explored. By itself, however, ICM did not perform well; a novel two-stage inference algorithm, with ICM followed by MCMCDA, fared much better, both in simulation and on real-world data.

The downside of the ICM-MCMC inference procedure is that very few guarantees can be made, since ICM is itself approximate and only reaches a local optimum. Additionally, even though the second-stage MCMCDA provides samples, they are not true samples from the full posterior, since the 'data points' it is trying to connect are in fact clusters found by ICM. Nevertheless, the idea of splitting the inference into within-epoch and between-epoch stages is appealing. The between-epoch stage of joining clusters into tracks also seems to have connections with split-merge methods (e.g., Jain and Neal, 2004). A two-stage sampling procedure that relies on the same intuition, but produces samples from the true posterior, should be within reach.

The inference algorithms presented in this chapter, and other traditional tracking algorithms such as MHT, all consider each view in sequence, sampling/scoring correspondence vectors given the associations from all previous views, but not future ones. That is, they are all performing *forward* filtering/sampling, but no smoothing is done in the space of associations. The previous chapter showed cases where this may be problematic. For the sampling-based algorithm in this chapter to be considered a true Gibbs sampler, it must condition on all information that is available to it, both past and future (if operating in batch mode, which is the case). The true Gibbs sampler for the DDPMM is currently being developed.

Finally, there is a serious theoretical issue with applying DPMMs, and by extension DDPMMs, to problems such as world modeling. The issue was raised recently by Miller and Harrison (2013, 2014); it is about the inconsistency in estimating the number of mixture components. In particular, Miller and Harrison (2013) showed a simple example where the posterior probability of the true number of components converged to 0 asymptotically (i.e., the true answer has zero probability!). This is problematic for object-based world modeling, since we are typically interested in the number of objects in the world. The issue is not a failure in the DP model; indeed, the DPMM was never intended to estimate the true number of components. Instead, it was originally conceived as a *density estimation* tool, for which the DPMM is consistent: the mixture density found in the DPMM converges to the true data-generating density. However, it tends to overestimate the number of mixture components; during inference, frequently samples have extra spurious clusters with few elements. Miller (2014) and Miller and Harrison (2015) suggested using an alternative *mixture of finite mixtures* (MFM) model, for which the inferred number of components is a consistent estimate. MFMs appear to be a promising alternative to DPMMs, and is likely generalizable to DDPMMs.

4.A Appendix: Derivation of posterior and predictive distributions for cluster parameters

In this appendix, I derive closed-form expressions for the posterior and predictive distributions of the parameter $\theta = (a, x)$, under the assumptions specified in Section 4.2.2.

The expressions for the fixed attribute are the same as in the previous chapter, since it is static. For convenience, I reproduce the equations here. Given a set of observations $\{b\}$:

$$\varphi(a) \triangleq \mathbb{P}\left(a \,|\, \{b\}\right) \propto \mathbb{P}\left(\{b\} \,|\, a\right)\ \mathbb{P}\left(a\right) \propto \left[\prod_{b_i \in \{b\}} \phi^a(b_i)\right]\ \pi(a) \tag{4.18}$$

$$\mathbb{P}\left(b' \,\middle|\, \{b\}\right) \propto \sum_a \mathbb{P}\left(b' \,\middle|\, a\right)\ \mathbb{P}\left(a \,|\, \{b\}\right) = \sum_a \phi^a(b')\ \varphi(a) \tag{4.19}$$

Given a set of observations $\left\{\left\{y_i^t\right\}_{i=1}^{N^t}\right\}_{t=\xi}^{\zeta}$ of the dynamic attributes, we can find the posterior distribution on $\left\{x^t\right\}_{t=\xi}^{\zeta}$ by performing Kalman filtering and smoothing (Kalman, 1960; Rauch et al., 1965). Applying a generic Kalman filter to the world modeling problem gives the following recursive filtering equations for $\left(\tilde{\mu}, \tilde{\Sigma}\right)$, the hyperparameters in the forward direction (during filtering):

$$\begin{aligned}
&\hat{\mu}^t = \tilde{\mu}^{t-1}\ , && \hat{\Sigma}^t = \tilde{\Sigma}^{t-1} + R(a) \\
&K^t = \begin{cases} \hat{\Sigma}^t \left(\hat{\Sigma}^t + \frac{S}{N^t}\right)^{-1}\ , & N^t > 0 \\ \mathbf{0}\ , & N^t = 0 \end{cases} \\
&\tilde{\mu}^t = \hat{\mu}^t + K^t\left(\bar{y}^t - \hat{\mu}^t\right)\ , && \tilde{\Sigma}^t = \left(I - K^t\right)\hat{\Sigma}^t
\end{aligned} \tag{4.20}$$

Recall that $R(a)$ is the covariance per time step of the random walk on x, and S is the covariance of the measurement noise distribution. For simplicity, I assume that the most-likely attribute value a is used in filtering; otherwise, we must marginalize over the posterior distribution $\varphi(a)$, which will lead to a weighted sum of covariance matrices that is difficult to handle if the covariances differ between the plausible attributes. The "$\hat{}$" variables are the predicted parameters before incorporating observations, and the "$\tilde{}$" variables are the parameters after incorporating observations (i.e., the Kalman filter output). Since there may be multiple observations of the pose in a single epoch, we have used an equivalent formulation involving the sample means $\bar{y}$, by exploiting the fact that if each $y_i^t \sim \mathcal{N}\left(x^t, S\right)$, then the sample mean has distribution $\bar{y}^t \sim \mathcal{N}\left(x^t, \frac{S}{N^t}\right)$. There may also be no observations at a given time, in which case the correction step has no effect ($K^t = 0$).

The Kalman filter is initialized with a noninformative prior:

$$\mu^0 = \mathbf{0}\ ,\ \Sigma^0 = \infty I \tag{4.21}$$

In practice, this implies that after the initial measurement(s) at time ξ, $\tilde{x}^\xi \sim \mathcal{N}\left(\bar{y}^\xi, \frac{S}{N^\xi}\right)$. To see

this, we can apply Equation 4.20 on $\left(\mu^0, \Sigma^0\right)$:

$$K^\xi = \left(\Sigma^0 + R(a)\right)\left(\Sigma^0 + R(a) + \frac{S}{N^\xi}\right)^{-1} \tag{4.22}$$

$$\tilde{\mu}^\xi = \mu^0 + K^\xi\left(\bar{y}^\xi - \mu^0\right) = K^\xi \bar{y}^\xi \tag{4.23}$$

$$\tilde{\Sigma}^\xi = \left(I - K^\xi\right)\left(\Sigma^0 + R(a)\right) \tag{4.24}$$

To handle the infinite initial covariance, we interpret Σ^0 as $\lim_{n\to\infty} nI$. This leads to:

$$\begin{aligned} K^\xi &= \lim_{n\to\infty}\left[nI\left(nI + R(a) + \frac{S}{N^\xi}\right)^{-1} + R(a)\left(nI + R(a) + \frac{S}{N^\xi}\right)^{-1}\right] \\ &= \lim_{n\to\infty}\left[\left(I + \frac{R(a)}{n} + \frac{1}{n}\frac{S}{N^\xi}\right)^{-1} + \frac{1}{n}R(a)\left(I + \frac{R(a)}{n} + \frac{1}{n}\frac{S}{N^\xi}\right)^{-1}\right] \\ &= I + 0 \cdot R(a) \cdot I = I \end{aligned} \tag{4.25}$$

Hence $\tilde{\mu}^\xi = K^\xi \bar{y}^\xi = \bar{y}^\xi$. For the covariance:

$$\begin{aligned} \tilde{\Sigma}^\xi &= \left[I - \left(\Sigma^0 + R(a)\right)\left(\Sigma^0 + R(a) + \frac{S}{N^\xi}\right)^{-1}\right]\left(\Sigma^0 + R(a)\right) \\ &= \left[\left(\Sigma^0 + R(a) + \frac{S}{N^\xi}\right)\left(\Sigma^0 + R(a) + \frac{S}{N^\xi}\right)^{-1} - \left(\Sigma^0 + R(a)\right)\left(\Sigma^0 + R(a) + \frac{S}{N^\xi}\right)^{-1}\right]\left(\Sigma^0 + R(a)\right) \\ &= \frac{S}{N^\xi}\left(\Sigma^0 + R(a) + \frac{S}{N^\xi}\right)^{-1}\left(\Sigma^0 + R(a)\right) \\ &= \lim_{n\to\infty}\left[\frac{S}{N^\xi}\left(nI + R(a) + \frac{S}{N^\xi}\right)^{-1} nI + \frac{S}{N^\xi}\left(nI + R(a) + \frac{S}{N^\xi}\right)^{-1} R(a)\right] \\ &= \lim_{n\to\infty}\left[\frac{S}{N^\xi}\left(I + \frac{R(a)}{n} + \frac{1}{n}\frac{S}{N^\xi}\right)^{-1} + \frac{1}{n}\frac{S}{N^\xi}\left(I + \frac{R(a)}{n} + \frac{1}{n}\frac{S}{N^\xi}\right)^{-1} R(a)\right] \\ &= \frac{S}{N^\xi} \cdot I + 0 \cdot \frac{S}{N^\xi} \cdot I \cdot R(a) = \frac{S}{N^\xi} \end{aligned} \tag{4.26}$$

In summary, choosing $\left(\mu^0, \Sigma^0\right) = (\mathbf{0}, \infty I)$ is equivalent to initializing the Kalman filter with $\left(\mu^\xi, \Sigma^\xi\right) = \left(\bar{y}^\xi, \frac{S}{N^\xi}\right)$, and proceeding for times $\xi < t \leq \zeta$.

After proceeding forward in time, information from later observations should also be propagated *backward* in time via a smoothing operation, to correct earlier estimates made during forward filtering. For example, in our application, the Rauch-Tung-Striebel (RTS) smoother runs the following

recursive operations starting at the final time ζ (Rauch et al., 1965):

$$\mu^{\zeta} = \tilde{\mu}^{\zeta} \ , \ \Sigma^{\zeta} = \tilde{\Sigma}^{\zeta} \tag{4.27}$$

$$C^t = \tilde{\Sigma}^t \left(\hat{\Sigma}^{t+1}\right)^{-1} = \tilde{\Sigma}^t \left(\tilde{\Sigma}^t + R(a)\right)^{-1} \tag{4.28}$$

$$\mu^t = \tilde{\mu}^t + C^t \left(\mu^{t+1} - \hat{\mu}^{t+1}\right) = \tilde{\mu}^t + C^t \left(\mu^{t+1} - \tilde{\mu}^t\right) \tag{4.29}$$

$$\Sigma^t = \tilde{\Sigma}^t + C^t \left(\Sigma^{t+1} - \hat{\Sigma}^{t+1}\right) \left(C^t\right)^{\intercal} = \tilde{\Sigma}^t + C^t \left(\Sigma^{t+1} - \tilde{\Sigma}^t - R(a)\right) \left(C^t\right)^{\intercal} \tag{4.30}$$

Recall that "$\hat{}$" and "$\tilde{}$" variables are the predicted and filtered parameters respectively. Parameters without such modifications are smoothed.

Once the sequence of parameters $\left\{\mu^t, \Sigma^t\right\}_{t=\xi}^{\zeta}$ is inferred, we can use them to determine the log-likelihood of the observations (for scoring associations) and the predictive distributions (for determining cluster assignment in Gibbs sampling). We will repeatedly use the following fact:

$$x \sim \mathcal{N}(\mu, \Sigma) \ , \ y|x \sim \mathcal{N}(x, \Lambda) \ \Rightarrow \ y \sim \int \mathbb{P}(y \,|\, x) \ \mathbb{P}(x) \ \mathrm{d}x = \mathcal{N}(\mu, \Sigma + \Lambda) \tag{4.31}$$

For example, we know that $x^t \sim \mathcal{N}\left(\mu^t, \Sigma^t\right)$ (hyperparameters obtained from Kalman smoothing), and from our modeling assumptions, $y^t|x^t \sim \mathcal{N}\left(x^t, S\right)$. Hence the marginal distribution over the pose observation (marginalized over all possible latent poses x^t) is $y^t \sim \mathcal{N}\left(\mu^t, \Sigma^t + S\right)$. From this we can immediately find the marginal likelihood of the observed data:

$$\mathbb{P}\left(\left\{\left\{y_i^t\right\}_{i=1}^{N^t}\right\}_{t=\xi}^{\zeta}\right) = \prod_{t=\xi}^{\zeta} \prod_{i=1}^{N^t} \mathbb{P}\left(y_i^t\right) = \prod_{t=\xi}^{\zeta} \prod_{i=1}^{N^t} \mathcal{N}\left(y_i^t \,;\, \mu^t, \Sigma^t + S\right) \tag{4.32}$$

This likelihood expression is used to score potential association hypotheses, e.g., for MCMCDA.

We can now derive the conditional probability expressions in the Gibbs sampler, shown in Equation 4.10. In collapsed Gibbs sampling, each observation's predictive likelihood $\mathbb{P}\left(o_i^t \,\middle|\, O_{\setminus ti}^k\right)$ involves an integral over the latent parameters $\theta^{kt} = \left(a^k, x^{kt}\right)$ of the cluster. In forward sampling, assigning observation o_i^t to cluster k has three cases:

1. If cluster k exists and is instantiated (i.e., has other observations at time t assigned to it), the posterior distribution of the pose x^{kt} is $\mathcal{N}\left(\mu^{kt}, \Sigma^{kt}\right)$, and the posterior distribution of the fixed attribute is $\varphi(a^k)$. Thus the predictive distribution is:

$$\begin{aligned}
\mathbb{P}\left(o_i^t \,\middle|\, O_{\setminus ti}^k\right) &= \int \mathbb{P}\left(o_i^t \,\middle|\, \theta\right) \ \mathbb{P}\left(\theta \,\middle|\, O_{\setminus ti}^k\right) \ \mathrm{d}\theta \\
&= \left[\sum_{a^k} \mathbb{P}\left(b_i^t \,\middle|\, a^k\right) \ \varphi(a^k)\right] \ \int \mathbb{P}\left(y_i^t \,\middle|\, x^{kt}\right) \ \mathcal{N}\left(x^{kt} \,;\, \mu^{kt}, \Sigma^{kt}\right) \ \mathrm{d}x^{kt} \\
&= \left[\sum_{a^k} \phi^{a^k}(b_i^t) \ \varphi(a^k)\right] \ \mathcal{N}\left(y_i^t \,;\, \mu^{kt}, \Sigma^{kt} + S\right)
\end{aligned} \tag{4.33}$$

In the final line, we used Equations 4.19 and 4.31 to simplify the predictive distributions.

2. If cluster k exists, but it has not yet been instantiated, this implies that, in the forward case, that the time of the observation t is beyond the final observed time $\tau = \zeta$ associated with the cluster. Then instead of integrating over the posterior distribution of x^{kt}, which does not exist yet, we need to integrate over its *predictive* distribution. This can be found by propagating the prediction step in the Kalman filter, starting from the final time step's distribution $x^{k\tau} \sim \mathcal{N}\left(\mu^{k\tau}, \Sigma^{k\tau}\right)$ (again, the most-likely attribute a^k is assumed):

$$\mu^{kt} = \mu^{k\tau}\ ,\ \Sigma^{kt} = \Sigma^{k\tau} + (t-\tau)R(a^k) \tag{4.34}$$

This is precisely the 'generalized' transition distribution $\tilde{T}$ for the pose in the DDPMM. Hence $x^{kt} \sim \mathcal{N}\left(\mu^{k\tau}, \Sigma^{k\tau} + (t-\tau)R(a^k)\right)$, and by a derivation similar to Equation 4.33:

$$\mathbb{P}\left(o_i^t \,\middle|\, O^k_{\setminus ti}\right) = \left[\sum_{a^k} \phi^{a^k}(b_i^t)\ \varphi(a^k)\right]\ \mathcal{N}\left(y_i^t\,;\,\mu^{k\tau}, \Sigma^{k\tau} + (t-\tau)R(a^k) + S\right) \tag{4.35}$$

3. If cluster k does not exist (and $O^k_{\setminus ti} = \emptyset$), then we should use the base distribution $H(\theta) \triangleq \pi(a)\,\mathcal{N}\left(x\,;\,\mu^0, \Sigma^0\right)$ instead of the posterior distribution. Then:

$$\begin{aligned}
\mathbb{P}\left(o_i^t \,\middle|\, O^k_{\setminus ti}\right) = \mathbb{P}\left(o_i^t\right) &= \int \mathbb{P}\left(o_i^t \,\middle|\, \theta\right)\ H(\theta)\ \mathrm{d}\theta \\
&= \left[\sum_{a^k} \mathbb{P}\left(b_i^t \,\middle|\, a^k\right)\ \pi(a^k)\right]\ \int \mathbb{P}\left(y_i^t \,\middle|\, x^{kt}\right)\ \mathcal{N}\left(x^{kt}\,;\,\mu^0, \Sigma^0\right)\ \mathrm{d}x^{kt} \\
&= \left[\sum_{a^k} \phi^{a^k}(b_i^t)\ \pi(a^k)\right]\ \mathcal{N}\left(y_i^t\,;\,\mathbf{0}, \infty I\right)
\end{aligned} \tag{4.36}$$

However, this requires the evaluation of an improper normal distribution in the final term. Since the choice of this prior was motivated by an attempt to give all initial poses equal probability, the same effect can be achieved by using a uniform distribution over the total explored world volume. Thus, in practice during Gibbs sampling we evaluate the following:

$$\mathbb{P}\left(o_i^t \,\middle|\, O^k_{\setminus ti}\right) = \left[\sum_{a^k} \phi^{a^k}(b_i^t)\ \pi(a^k)\right]\ \mathrm{Unif}(\mathrm{vol}(\mathrm{world})) \tag{4.37}$$

This expression is similar to the expression for the observation likelihood of false positives. If the observation is actually assigned to a new cluster, then we revert to the noninformative normal prior and perform Kalman smoothing, which is now no longer problematic since it does not require evaluation of improper densities.

Chapter 5

Not Seeing is Also Believing: Combining Object and Metric Spatial Information

Spatial representations are fundamental to mobile robots operating in uncertain environments. A navigating mobile robot needs to know which places are free to move into and what obstacles it might collide with. A mobile-manipulation robot cooking at home needs to be able to find and detect objects such as kitchen utensils and ingredients. These two tasks typically represent space in distinct ways: navigation with occupancy grid maps, which I will refer to as 'metric-level'; mobile manipulation with objects and their attributes, which is 'object-level'. Many tasks represent space in just one of these two ways, use them in parallel without information flow, or infer one solely from the other, but rarely is there any interaction between the two levels.

Consider a motivating example, as depicted in Figure 5-1. A mobile robot with a camera mounted on top takes an image and sees the side of a shelf on a table. From the camera point cloud, it infers that a shelf is present, and estimates the shelf's pose, shown in red and indicated by the white arrow. Even though most of the shelf lies within an unobserved region of space, as indicated by the gray 'fog' on the right, the robot can infer that the space overlapping with the box at its estimated pose is occupied (by the shelf). This is an example of object-to-metric inference. Through the act of seeing the shelf, the robot also knows that the rays between its camera and the front of the shelf passed through free (unoccupied) space. Since this space is free, the robot can also infer that no objects are present in this space. This is an example of metric-to-object inference. We will consider more examples of both types of information interaction in this chapter.

With effort, it is typically possible to use only a single layer of spatial representation. However, this can unnecessarily complicate the storage of information and the updating of the representation, because certain types of information from sensors are more compatible with specific types of representation. An identified rigid object is inherently atomic, but this is not respected when treated as a collection of discretized grid cells. If the object is moved, then instead of simply updating a 'pose' attribute in the object state, the the entire collection of grid cells will need to be updated. Conversely, free space is easy to represent in an occupancy grid. However, because it provides in-

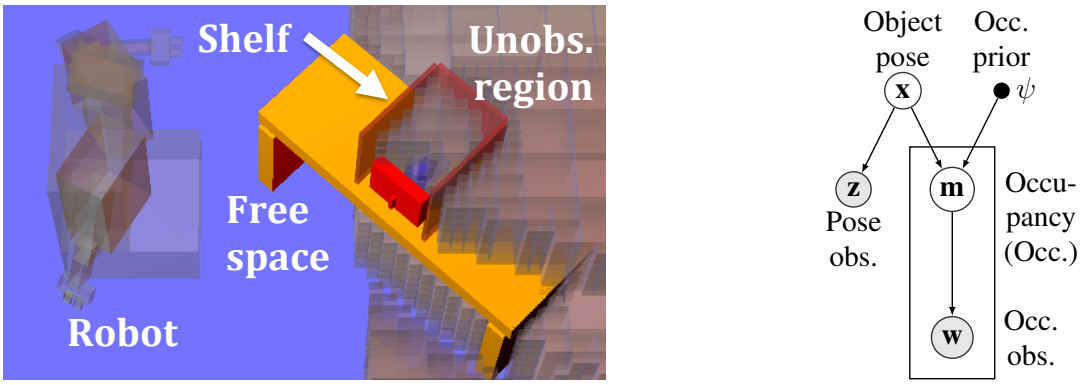


Figure 5-1: A mobile robot uses object detections to infer regions of occupied space, and uses free space observations to eliminate possible locations of objects. The framework in this chpater allows inference across representational layers as depicted by the graphical model; see Sections 5.1–5.3 for details.

formation about the absence of any object, which amounts to 'cutting holes' in each object's pose distribution, forcing free space information to be kept in pose space leads to complicated pose distributions and updates that scale with the number of known objects instead of the number of newly observed cells. Moreover, much of this complex updating is wasted, because information about a local region of free space would not affect an object's pose unless the object is nearby.

Our goal is to combine the advantages of each layer of representation and provide a framework for integrating both types of information. In particular, I adopt the philosophy of keeping each type of information in its 'natural' representation, where it can be easily updated, and *only* combining them when queries about specific states are made. This is an efficiency trade-off between filtering and querying; we strive for simplicity and compactness in the former by delaying computation to query-time. The specific representational choices made will be explored in greater detail next.

To illustrate our strategy, I first show, in detail, the approach for a concrete one-dimensional discrete world involving a single object. At that point, the general case is in fact not too different. Through several illustrative example applications of the framework, I will demonstrate, for example, how free space information can be used to reduce uncertainty in object type and pose, and why object-level representations are necessary to maintain an accurate metric spatial representation. A quantitative comparison between our framework and a generic particle filter is also given. Finally, further justification for our strategy is provided through a high-level graphical analysis.

5.1 Problem Definition and Solution Strategy

Consider a well-localized robot making observations in a world containing stationary objects. Since the contents of a spatial representation is ultimately a state estimate, I first describe the state. I assume that each object obj^i is described by a fixed set of attributes of interest, whose values are concatenated into a vector $\mathbf{x}^i$. Likewise, the world is discretized into a metric grid (not necessarily evenly spaced), where each cell $cell^j$ is endowed with another set of attributes with value $\mathbf{m}^j$. For

concreteness, it may help to consider $\mathbf{x}^i$ being the object pose (assuming we know which object it is), and $\mathbf{m}^j$ being the binary occupancy value for $cell^j$. I shall explore this case further in Section 5.2, and subsequently generalize to other attributes in Section 5.3.

The objects' states $\{\mathbf{x}^i\}$ and the cells' states $\{\mathbf{m}^j\}$ are not known, and are imperfectly sensed by the robot. I assume that the perception framework returns two independent types of observations, $\{\mathbf{z}^i_{1:Z}\}$ and $\{\mathbf{w}^j_{1:W}\}$, revealing information about the objects and cells respectively. The subscripts indicate that each object/cell may have multiple observations. Observations may be raw sensor readings or be the output of some intermediate perception pipeline. For example, $\mathbf{w}^j$ may be range sensor readings, whereas $\mathbf{z}^i$ may be the output of an object detection and pose estimation pipeline.

For convenience, I will use the following shorthand in the rest of the chapter. As in the above presentation, superscripts always refer to the index of the object/cell. To avoid the clutter of set notation, we will denote the set of all objects' states, $\{\mathbf{x}^i\}$, by $\mathbf{x}^\bullet$; specific indices will denote individual states (e.g., $\mathbf{x}^i$ is obj^i's state). Similarly, $\mathbf{m}^j$ is $cell^j$'s state, whereas $\mathbf{m}^\bullet$ refers to the states of all cells (previously $\{\mathbf{m}^j\}$). Likewise, for observations, $\mathbf{z}^i_k$ is the k'th observation associated with obj^i, $\mathbf{z}^i_\bullet$ is the set of observations associated with obj^i, and $\mathbf{z}^\bullet$ is the set of all object observations (previously $\{\mathbf{z}^i_{1:Z}\}$).

Our goal is to estimate the marginal posterior distributions:

$$\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet) \quad \text{and} \quad \mathbb{P}(\mathbf{m}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet) \tag{5.1}$$

In most of our examples, such as the object-pose/cell-occupancy one described above, $\mathbf{x}^\bullet$ and $\mathbf{m}^\bullet$ are dependent: given that an object is in pose x, the cells that overlap with the object at pose x must be occupied. Object-based dependencies also tend to be local to the space that the object occupies and hence very non-uniform: cells that do *not* overlap with the object are essentially unaffected. The lack of uniformity dashes all hopes of a nice parametric update to the objects' states. For example, if m^j is known to be free, all poses that overlap m^j must have zero probability, thereby creating a 'hole' in pose space that is impossible to represent using a typical Gaussian pose distribution.

As a result, we must resort to non-parametric representations, such as a collection of samples, to achieve good approximations to the posterior distributions. However, the dimension of the joint state grows with the number of objects and the size of the world, and sampling in the joint state quickly becomes intractable in any realistic environment. This approach can be made feasible with aggressive factoring of the state space; however, combining different factors correctly simply introduces another fusion problem. Filtering a collection of samples over time, or particle filtering (Doucet et al., 2001; Thrun et al., 2001), also introduces particle-set maintenance issues.

Instead of filtering in the joint state and handling complex dependencies, our strategy is to filter *separately* in the object and metric spaces, and *merge* them on demand as queries about either posterior are made. Our philosophy is to trade off filter accuracy for runtime efficiency (by using more restrictive representations that each require ignoring different parts of the perceived data), while ensuring that appropriate corrections are made when answering queries. By making typical independence assumptions within each layer, we can leverage standard representations such as a

Kalman filter (for object pose) and an occupancy grid (for metric occupancy) to make filtering efficient. Specifically, we propose to maintain the following distributions in two filters:

$$\mathbb{P}(\mathbf{x}^\bullet \,|\, \mathbf{z}^\bullet) \quad \text{and} \quad \mathbb{P}(\mathbf{m}^\bullet \,|\, \mathbf{w}^\bullet) \tag{5.2}$$

We *only* incorporate the other source of information at query time. Computing the posteriors in Equation 5.1 from the filtered distributions in Equation 5.2 is the subject of the next section.

5.2 The One-Dimensional, Single-Object Case

To ground our discussion of the solution strategy, in this section we consider a simple instance of the problem discussed in the previous section. In particular, we focus on estimating the (discrete) location of a single static object and the occupancy of grid cells in a discretized one-dimensional world. The general problem involving more objects and other attributes is addressed later.

5.2.1 Formulation

The single-object, 1-D instance is defined as follows:

- The 1-D world consists of C contiguous, unit-width cells with indices $1 \leq j \leq C$.
- A static object of interest, with known length L, exists in the world. Its location, the lowest cell index it occupies, is the only attribute being estimated. Hence its state $\mathbf{x}$ satisfies $\mathbf{x} \in [1, C - L + 1] \triangleq \{1, \ldots, C - L + 1\}$.
- We are also interested in estimating the occupancy of each cell $cell^j$. Each cell's state $\mathbf{m}^j$ is binary, with value 1 if it is occupied and 0 if it is free.
- Cells may be occupied by the object, occupied by 'dirt'/'stuff', or be free. 'Stuff' refers to physically-existing entities that we either cannot yet or choose not to identify. Imagine only seeing the tip of a handle (which makes the object difficult to identify) or, as the name suggests, a ball of dirt (which we choose to ignore except note its presence). We will not explicitly distinguish between the two types of occupancy; the cell's state has value 1 if it is occupied by either the object or 'stuff', and 0 if it is free.
- The assumption above, that cells can be occupied by non-object entities, allows us to ascribe a simple prior model of occupancy: each cell is occupied independently with known probability $\mathbb{P}(\mathbf{m}^j = 1) = \psi$. This prior model and cell independence assumption are commonly used in the occupancy grid literature (Thrun et al., 2005). This may be inaccurate, especially if the object is long and ψ is small.
- Noisy observations $\mathbf{z}_\bullet$ of the single object's location $\mathbf{x}$ and observations $\mathbf{w}^\bullet$ of the cells' occupancies $\mathbf{m}^\bullet$ are made. We will be intentionally agnostic to the specific sensor model used, and only assume that appropriate filters are used in light of the noise models.

- The object and metric filters maintain $\mathbb{P}(\mathbf{x} \mid \mathbf{z}_\bullet)$ and $\mathbb{P}(\mathbf{m}^\bullet \mid \mathbf{w}^\bullet)$ respectively. I assume that the former is a discrete distribution over the domain of $\mathbf{x}$, and the latter is an occupancy grid, using the standard log-odds ratio $\ell^j = \log \frac{\mathbb{P}(\mathbf{m}^j=1 \mid \mathbf{w}^j_\bullet)}{\mathbb{P}(\mathbf{m}^j=0 \mid \mathbf{w}^j_\bullet)}$ for each cell's occupancy.
- States of distinct cells are assumed to be conditionally independent given the object state $\mathbf{x}$. This is a relaxation of the assumption cells are independent, which is typically assumed in occupancy grids. The current assumption disallows arbitrary dependencies between cells; only dependencies mediated by objects are allowed. For example, two adjacent cells may be occupied by the same object and hence are dependent if the object's location is not known.

As mentioned in the previous section, what makes this problem interesting is that $\mathbf{x}$ and $\mathbf{m}^\bullet$ are dependent. In this case, the crucial link is that an object that is located at $\mathbf{x}$ *necessarily occupies* cells with indices $j \in \mathcal{J}(\mathbf{x}) \triangleq [\mathbf{x}, \mathbf{x} + L - 1]$, and therefore these cells must have as state $\mathbf{m}^j = 1$. This means that states of a subset of cells are strongly dependent on the object state, and we expect this to appear in the metric posterior $\mathbb{P}(\mathbf{m}^\bullet \mid \mathbf{z}_\bullet, \mathbf{w}^\bullet)$. Likewise, occupancy/freeness of a cell also supports/opposes respectively the hypothesis that an object overlaps the cell. However, the latter dependency is weaker than the former one, as an occupied cell can be due to 'dirt' (or other objects, though not in this case), and a free cell typically only eliminates a small portion of the object location hypotheses.

5.2.2 Cell occupancy posterior

We now use this link between $\mathbf{x}$ and $\mathbf{m}^\bullet$ to derive the desired posterior distributions from Equation 5.1. First consider the posterior occupancy $\mathbf{m}^j$ of a single cell $cell^j$. Intuitively, we expect that if the object likely overlaps $cell^j$, the posterior occupancy should be close to 1, whereas if the object is unlikely to overlap the cell, then the posterior occupancy should be dictated by the 'stuff' prior and relevant occupancy observations ($\mathbf{w}^j_\bullet$). Since we do not know the exact location of the object, we instead have to consider all possibilities:

$$\mathbb{P}(\mathbf{m}^j \mid \mathbf{z}_\bullet, \mathbf{w}^\bullet) = \sum_{\mathbf{x}} \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}, \mathbf{w}^j_\bullet)\, \mathbb{P}(\mathbf{x} \mid \mathbf{z}_\bullet, \mathbf{w}^\bullet) \tag{5.3}$$

In the first term, because $\mathbf{x}$ is now explicitly considered, object observations $\mathbf{z}_\bullet$ are no longer informative and are dropped. Since we assumed that cells are conditionally independent given the object state, all other cells' observations are dropped too. The second term is the posterior distribution on the object location, which will be discussed later.

The term $\mathbb{P}(\mathbf{m}^j \mid \mathbf{x}, \mathbf{w}^j_\bullet)$ can be decomposed further:

$$\mathbb{P}(\mathbf{m}^j \mid \mathbf{x}, \mathbf{w}^j_\bullet) \propto \mathbb{P}(\mathbf{w}^j_\bullet \mid \mathbf{m}^j)\, \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}) \tag{5.4}$$

The second term, $\mathbb{P}(\mathbf{m}^j \mid \mathbf{x})$, serves as the link between cells and objects. By the discussion above, for $j \in \mathcal{J}(x)$, i.e., cells that the object at location $\mathbf{x}$ overlaps, $\mathbf{m}^j$ must be 1. In this case, Equation 5.4 is only non-zero for $\mathbf{m}^j = 1$, so $\mathbb{P}(\mathbf{m}^j = 1 \mid \mathbf{x}, \mathbf{w}^j_\bullet)$ must also be 1. For $j \notin \mathcal{J}(x)$, the cell is unaffected

by the object, hence $\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{x}) = \mathbb{P}(\mathbf{m}^j)$. Equation 5.4 in this case is, by reverse application of Bayes' rule, proportional to $\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{w}_\bullet^j)$, and since this is in fact a distribution, $\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{x}, \mathbf{w}_\bullet^j) = \mathbb{P}(\mathbf{m}^j \,|\, \mathbf{w}_\bullet^j)$. This reflects that for $j \notin \mathcal{J}(x)$, the cell's state is independent of the object state. In summary:

$$\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{x}, \mathbf{w}_\bullet^j) = \begin{cases} 1\,, & \text{if } j \in \mathcal{J}(\mathbf{x}) \\ \mathbb{P}(\mathbf{m}^j \,|\, \mathbf{w}_\bullet^j)\,, & \text{otherwise} \end{cases} \tag{5.5}$$

This 'link' between object and cell states matches the intuition given above: the cell is necessarily occupied if the object overlaps it; otherwise, the object state is ignored and only occupancy observations are used. The probability value $\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{w}_\bullet^j)$ is readily available from the metric filter (for an occupancy grid with log-odds ratio ℓ^j for $cell^j$, the desired probability is $1 - \frac{1}{1+\exp(\ell^j)}$). Combining Equations 5.3 and 5.5 results in a nicely interpretable posterior:

$$\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet) = p_{\text{overlap}} + \mathbb{P}(\mathbf{m}^j \,|\, \mathbf{w}_\bullet^j)\,(1 - p_{\text{overlap}}) \tag{5.6}$$

Let $p_{\text{overlap}} \triangleq \mathbb{P}(\mathbf{x} \in [j - L + 1, j] \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet)$, the posterior probability that the object is in a location that overlaps $cell^j$. To compute this value, we need the object location's posterior distribution.

5.2.3 Object location posterior

By Bayes' rule,

$$\mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet) \propto \mathbb{P}(\mathbf{w}^\bullet \,|\, \mathbf{x}, \mathbf{z}_\bullet)\,\mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet) = \mathbb{P}(\mathbf{w}^\bullet \,|\, \mathbf{x})\,\mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet) \tag{5.7}$$

The second term is maintained by the object filter, and in this context acts as the 'prior' of the object location given only object-level observations. This distribution is adjusted by the first term, which weighs in the likelihood of occupancy observations. To evaluate this, we need to consider the latent cell occupancies $\mathbf{m}^\bullet$, and the constraint imposed by $\mathbf{x}$. Once again, cells overlapping the object must be occupied ($\mathbf{m}^j = 1$), so we only need to consider possibilities for the other cells. The non-overlapping cells are independent of $\mathbf{x}$, and are occupied according to the prior model (independently with probability ψ). Hence:

$$\begin{aligned} \mathbb{P}(\mathbf{w}^\bullet \,|\, \mathbf{x}) &= \sum_{\mathbf{m}^\bullet} \mathbb{P}(\mathbf{w}^\bullet, \mathbf{m}^\bullet \,|\, \mathbf{x}) = \left[\prod_{j \notin \mathcal{J}(\mathbf{x})} \sum_{\mathbf{m}^j=0}^{1} \mathbb{P}(\mathbf{w}_\bullet^j \,|\, \mathbf{m}^j)\,\mathbb{P}(\mathbf{m}^j)\right] \left[\prod_{j \in \mathcal{J}(\mathbf{x})} \mathbb{P}(\mathbf{w}_\bullet^j \,|\, \mathbf{m}^j = 1)\right] \\ &= \left[\prod_j \mathbb{P}(\mathbf{w}_\bullet^j)\right] \left[\prod_{j \notin \mathcal{J}(\mathbf{x})} \sum_{\mathbf{m}^j} \mathbb{P}(\mathbf{m}^j \,|\, \mathbf{w}_\bullet^j)\right] \left[\prod_{j \in \mathcal{J}(\mathbf{x})} \frac{\mathbb{P}(\mathbf{m}^j = 1 \,|\, \mathbf{w}_\bullet^j)}{\mathbb{P}(\mathbf{m}^j = 1)}\right] \\ &= \eta(\mathbf{w}^\bullet) \times 1 \times \left[\prod_{j \in \mathcal{J}(\mathbf{x})} \frac{1}{\psi}\left(1 - \frac{1}{1 + \exp(\ell^j)}\right)\right] \end{aligned} \tag{5.8}$$

In the second line we used the conditional independence of cell states given $\mathbf{x}$ to factor the expression, and $\eta(\mathbf{w}^\bullet)$ represents the first product in the penultimate line.

When substituting Equation 5.8 back into Equation 5.7, recall that since $\mathbf{w}^\bullet$ is given, and we only need $\mathbb{P}(\mathbf{w}^\bullet \,|\, \mathbf{x})$ up to proportionality, we can ignore the $\eta(\mathbf{w}^\bullet)$ term. Hence:

$$\mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet) \propto \left[\prod_{j \in \mathcal{J}(\mathbf{x})} \frac{1}{\psi} \left(1 - \frac{1}{1 + \exp(\ell^j)} \right) \right] \mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet) \tag{5.9}$$

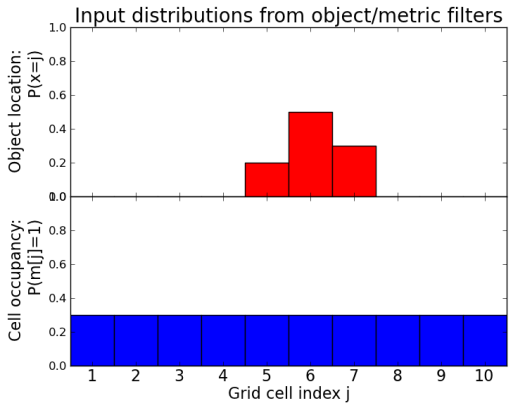
Note that the expression above only contains $O(L)$ terms, since $\mathcal{J}(x)$ contains exactly L cells. The complexity therefore scales with the number of cells the object affects, instead of with the whole world (containing C cells, which is potentially much greater than L). For discrete $\mathbf{x}$ with X possible states, computing $\mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet)$ therefore requires $O(LX)$ time, since Equation 5.7 must be normalized over all possible $\mathbf{x}$. Finally, we have all the pieces needed to compute $\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet)$ as well using Equation 5.6. To compute both the object and metric posterior distributions, we first find the former using Equation 5.9, then find the posterior occupancy of each cell using Equation 5.6. This procedure requires $O(LX + C)$ time. In practice, when operating in local regions of large worlds, it is unlikely that one would want the posterior occupancy of all cells in the world; only cells of interest need to have their posterior state computed.

5.2.4 Demonstrations

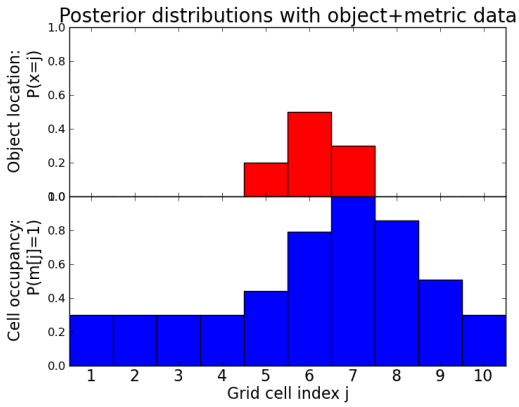
To illustrate the above approach, we consider two simple examples where the world contains $C = 10$ cells and a single object of length $L = 3$. In each case, only one type of information (object location or cells' occupancy) has been observed. The methods described in this section are used to propagate the information to the other representation.

In Figure 5-2, we consider the case when only object locations have been observed. Figure 5-2(a) show distributions obtained from object (top) and metric (bottom) filters, i.e., $\mathbb{P}(\mathbf{x} \,|\, \mathbf{z}_\bullet)$ and $\mathbb{P}(\mathbf{m}^\bullet \,|\, \mathbf{w}^\bullet)$ respectively. The object filter contains a *single* distribution, so the top plot sums to 1, whereas the metric filter contains a *collection* of binary distributions, one for each cell, so the bottom plot does *not* sum to 1. The object filter determines that the object can only be located at cells 5–7 (recall that this is the left-most point of the object). No occupancy observations have been made, so each cell's occupancy probability is initially the prior value, $\psi = 0.3$. In Figure 5-2(b) after applying our framework, the posterior occupancy distribution $\mathbb{P}(\mathbf{m}^\bullet \,|\, \mathbf{z}_\bullet, \mathbf{w}^\bullet)$ reflects the fact that cells 5–9 might be occupied by the object, even though *no occupancy measurements have been made.* In particular, all possibilities of the object location require *cell*[7] to be occupied, hence its occupancy probability is 1. Cells with no possible overlap with the object are left unchanged. The distribution on object location is unchanged since there are no additional observations to be considered.

In Figure 5-3, only occupancies of some cells have been observed. Cells 5–7 have many observations indicating that they are free, and cell 4 had only one observation indicating that it is occupied. No object observations have been made, so the object location distribution is uniform



(a) Filter distributions (input)



(b) Posterior distributions (output)

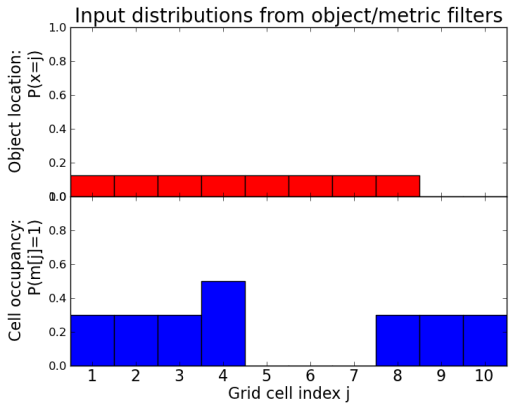
Figure 5-2: Using only object observations, the object filter maintains a distribution over the object's locations (top left). The object filter contains a *single* distribution, so top plots each sums to 1, whereas the metric filter contains a *collection* of binary distributions, one for each cell, so bottom plots do *not* sum to 1. Some cells have increased posterior probability of occupancy (bottom right), even though *no occupancy observations have been made*. Please see text in Section 5.2.4 for details.

over the feasible range. The free cells in the middle of the location posterior distribution (top right) indicate that it is highly unlikely that any object can occupy those cells (which correspond to $\mathbf{x} \in [3, 7]$). This makes the posterior distribution multi-modal. Also, the weak evidence that $cell^4$ is occupied gives a slight preference for $\mathbf{x} = 2$. Again, even though *the object has never been observed*, the posterior distribution on its location is drastically narrowed! Unlike the previous case, the occupancy distribution has changed by virtue of the domain assumption that an object must exist. Unobserved cells are also affected by this process; in fact, $cell^2$ and $cell^3$ are now *even more* likely to be occupied than $cell^4$ (which had the only observation of being occupied) because of the possibility that $\mathbf{x} = 1$.

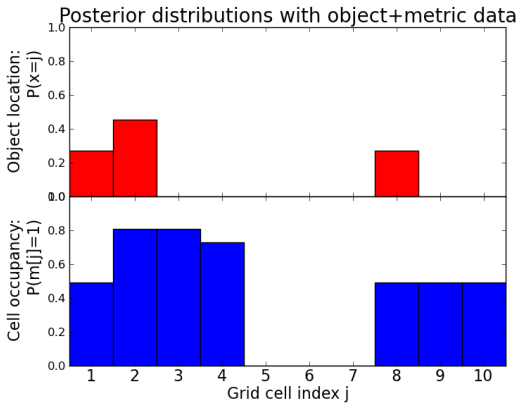
5.3 Generalizing to Arbitrary States

The previous section used several concrete simplifications: the world was one-dimensional, exactly one object existed in the world, the object's shape (length) was given, and the only attributes considered were object location and cell occupancy. I will remove all these simplifications in this section. I will also discuss a way of handling continuous object states at the end.

Despite removing many simplifications, the conceptual framework for computing the two desired posterior distributions is actually quite similar to the development in the previous section. The major differences now are that multiple objects are present ($\mathbf{x}^\bullet, \mathbf{z}^\bullet$ instead of $\mathbf{x}, \mathbf{z}$), and that domain-specific derivations are no longer applicable in general. We still require the core representational assumption that an object-based filter and a metric-based filter are maintained to provide efficient access to $\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet)$ and $\mathbb{P}(\mathbf{m}^\bullet \mid \mathbf{w}^\bullet)$ respectively. The latter will typically be maintained



(a) Filter distributions (input)



(b) Posterior distributions (output)

Figure 5-3: Using only cell occupancy/freeness observations, the posterior of the object's location is changed drastically even though *the object has never been observed.* See text in Section 5.2.4 for details.

independently for each cell, with distribution $\mathbb{P}(\mathbf{m}^j \mid \mathbf{w}_\bullet^j)$ for $cell^j$. The typical grid cell assumptions are retained as well: cell states are conditionally independent given all object states, and states have a known prior distribution $\mathbb{P}(\mathbf{m}^j)$.

Following the derivation in Equations 5.3 and 5.4, we get for $cell^j$'s posterior distribution:

$$\mathbb{P}(\mathbf{m}^j \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet) = \sum_{\mathbf{x}^\bullet} \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet, \mathbf{w}_\bullet^j)\, \mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet)\,, \text{ where} \tag{5.10}$$

$$\mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet, \mathbf{w}_\bullet^j) \propto \mathbb{P}(\mathbf{w}_\bullet^j \mid \mathbf{m}^j)\, \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet) \propto \frac{\mathbb{P}(\mathbf{m}^j \mid \mathbf{w}_\bullet^j)}{\mathbb{P}(\mathbf{m}^j)}\, \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet) \tag{5.11}$$

Again assuming that we have already computed the posterior object state $\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet)$, all other terms are given except for $\mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet)$. This distribution is the fundamental link between cells and objects, specifying in a generative fashion how objects' states affect each cell's state (which can be considered individually since cell states are conditionally independent given $\mathbf{x}^\bullet$). We will see other examples of this linking distribution in the next section.

For the posterior distribution on object states, we can likewise follow the derivation in Equations 5.7 and 5.8:

$$\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet) \propto \mathbb{P}(\mathbf{w}^\bullet \mid \mathbf{x}^\bullet)\, \mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet), \text{ where} \tag{5.12}$$

$$\begin{aligned} \mathbb{P}(\mathbf{w}^\bullet \mid \mathbf{x}^\bullet) &= \sum_{\mathbf{m}^\bullet} \mathbb{P}(\mathbf{w}^\bullet \mid \mathbf{m}^\bullet)\, \mathbb{P}(\mathbf{m}^\bullet \mid \mathbf{x}^\bullet) \\ &= \sum_{\mathbf{m}^\bullet} \left[\prod_j \mathbb{P}(\mathbf{w}_\bullet^j \mid \mathbf{m}^j)\right] \left[\prod_j \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet)\right] \\ &\propto \prod_j \left[\sum_{\mathbf{m}^j} \frac{\mathbb{P}(\mathbf{m}^j \mid \mathbf{w}_\bullet^j)}{\mathbb{P}(\mathbf{m}^j)}\, \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet)\right] \end{aligned} \tag{5.13}$$

Again, all terms needed to compute the above are available from the filters, the cell prior, and the object-cell link $\mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet)$ described earlier.

As in the previous section, we can compute this latter posterior distribution more efficiently by considering only the cells that objects affect. For any particular assignment to $\mathbf{x}^\bullet$, let $\mathcal{J}(\mathbf{x}^\bullet)$ be defined to be the indices of cells whose state $\mathbf{m}^j$ depends on $\mathbf{x}^\bullet$. This implies that if $j \notin \mathcal{J}(\mathbf{x}^\bullet)$, then $\mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet) = \mathbb{P}(\mathbf{m}^j)$, and their respective terms in the product of Equation 5.13 are independent of $\mathbf{x}^\bullet$. In fact, for $j \notin \mathcal{J}(\mathbf{x}^\bullet)$, the sum is equal to 1, a consequence of the fact that $\mathbb{P}(\mathbf{w}^j_\bullet \mid \mathbf{x}^\bullet) = \mathbb{P}(\mathbf{w}^j_\bullet)$ in this case. Hence:

$$\begin{aligned}
&\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet) \\
&\quad \propto \left[\prod_{j \in \mathcal{J}(\mathbf{x}^\bullet)} \sum_{\mathbf{m}^j} \frac{\mathbb{P}(\mathbf{m}^j \mid \mathbf{w}^j_\bullet)}{\mathbb{P}(\mathbf{m}^j)} \mathbb{P}(\mathbf{m}^j \mid \mathbf{x}^\bullet) \right] \mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet)
\end{aligned} \tag{5.14}$$

Similar to Equation 5.9, the number of product terms has been reduced from the number of cells to $O(|\mathcal{J}(\mathbf{x}^\bullet)|)$, for each $\mathbf{x}^\bullet$. This is potentially a major reduction because objects, for each particular state they are in, may only affect a small number of cells (e.g., the ones they occupy). Unfortunately, the expression still scales with the domain size of $\mathbf{x}^\bullet$, which grows exponentially with the number of objects. In practice, approximations can be made by bounding the number of objects considered jointly and aggressively partitioning objects into subsets that are unlikely to interact with each other. Alternatively, sampling values of $\mathbf{x}^\bullet$ from the filter posterior $\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet)$ can produce good state candidates.

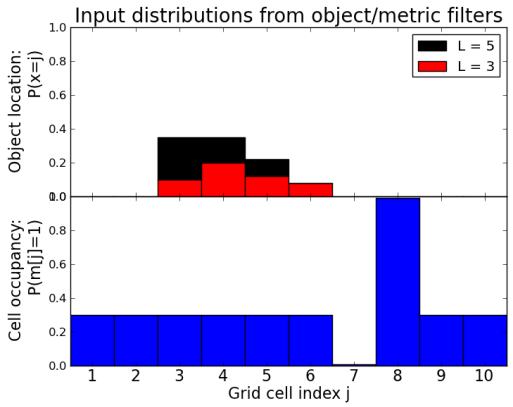
Object state attributes can be continuous, for example using Gaussian distributions to represent pose. However, the above framework can only handle discrete states. Apart from discretizing the state space, one can instead sample objects' states from the filter $\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet)$ and use Equation 5.14 to form an approximate posterior distribution, represented as a weighted collection of samples. These samples can then be used in Equation 5.10 to compute a Monte-Carlo estimate of $cell^j$'s posterior distribution $\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet, \mathbf{w}^\bullet)$. We will see an example of this in Section 5.5.

5.4 Applications

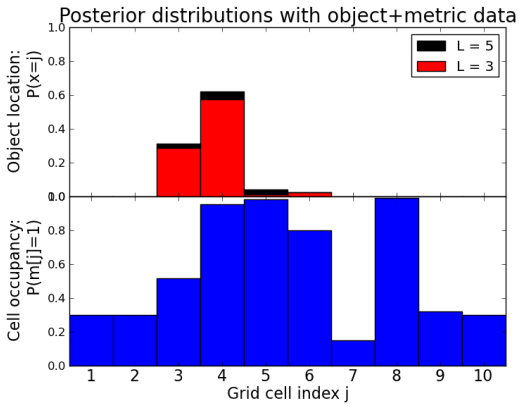
In this section, we will look at several illustrative scenarios where object-based and metric-based information need to be considered together. First, I will introduce additional attributes (besides location and occupancy from Section 5.2).

5.4.1 Shape-based object identification

When detecting and tracking objects in the world, uncertainty typically arises in more attributes than just location/pose. In particular, object recognition algorithms are prone to confusing object types, especially if we only have a limited view of the object of interest. When multiple instances of the same object type are present, we also run into data association issues, as discussed in the



(a) Filter distributions (input)



(b) Posterior distributions (output)

Figure 5-4: A 1-D scenario with a single object, but now the object's length is uncertain as well. The object filter (top left) determines that the object may have length $L = 3$ or 5, and for either case, may be in one of several locations. Because of a strong free space occupancy observation in $cell^7$, the uncertainty in object length has decreased significantly in the posterior object distribution (top right), because a $L = 5$ object must contradict the free space evidence of $cell^7$. See text in Section 5.4.1 for more details.

previous chapters. Furthermore, we may even be unsure about the number of objects in existence. Sophisticated filters (e.g., Hager and Wegbreit, 2011; Elfring et al., 2013; Wong et al., 2015) can maintain distributions over hypotheses of the world, where a hypothesis in our context is an assignment to the joint state $\mathbf{x}^\bullet$.

Let us revisit the one-dimensional model of Section 5.2 again, this time with uncertainty in the object type. In particular, the single object's length L is unknown, and is treated as an attribute in $\mathbf{x}^\bullet$ (in addition to the object's location). Suppose that after making some observations of the object, we get $\mathbb{P}(\mathbf{x}^\bullet \mid \mathbf{z}^\bullet)$ from the filter, as shown in Figure 5-4(a) (top). The filter has identified two possible lengths of the object ($L = 3, 5$). Here we visualize the two-dimensional distribution as a stacked histogram, where the bottom bars (red) shows the location distribution for $L = 3$, and the top bars (black) for $L = 5$. The total height of the bars is the marginal distribution of the object's location. Suppose we have also observed that $cell^7$ is most likely empty, and $cell^8$ most likely occupied (the occupancy grid gives probability of occupancy 0.01 and 0.99 for the two cells respectively). The posterior distributions obtained by combining the filters' distributions is shown on in Figure 5-4(b).

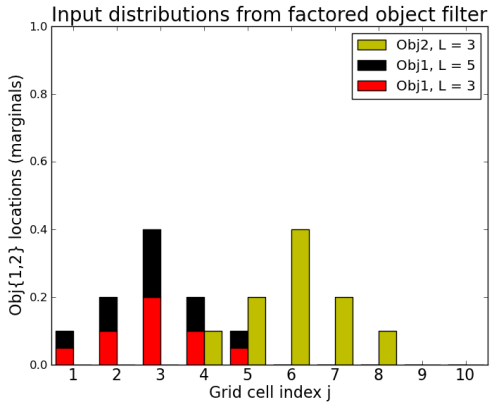
In the object-state posterior, the main difference is that the probability mass has shifted away from the $L = 5$ object type, and towards locations on the left. Both effects are caused by the free space observations of $cell^7$. Because all locations for $L = 5$ states cause the object to overlap $cell^7$, implying that $cell^7$ is occupied, the observations that suggest otherwise cause the marginal probability of $L = 5$ to drop from 0.50 to 0.10. The drop in probability for locations 5 and 6 is due to the same reason. In conclusion, incorporating occupancy information has allowed us to reduce the uncertainty in both object location *and* object type (length).

Interestingly, among the $L = 5$ states, although location 3 had the highest probability from the object filter, it has the lowest posterior probability mass. This minor effect comes from the likely occupancy of $cell^8$, which lends more evidence to the other $L = 5$ states (which overlap $cell^8$) but not for location 3 (which does not overlap). However, the strong evidence of $cell^8$'s occupancy has much less of an effect compared to the free space evidence of $cell^7$. This example highlights the fundamental asymmetry in occupancy information. Recall that the prior model allows for unidentified, non-object 'stuff' to exist in the world, stochastically with probability ψ. That $cell^8$ is occupied only suggests it is overlapped by some object, or contains 'stuff'. In particular, this is the interpretation for $cell^8$ for the two most likely $L = 3$ states in the posterior. An object overlapping the cell would gain evidence, but the cell's occupancy does not need to be explained by an object. In contrast, $cell^7$ being free means that *none* of the objects can overlap it, thereby enforcing a strong constraint on each object's state. In the example shown in Figure 5-4, this constraint allowed us to identify that the object is most likely of shorter length.

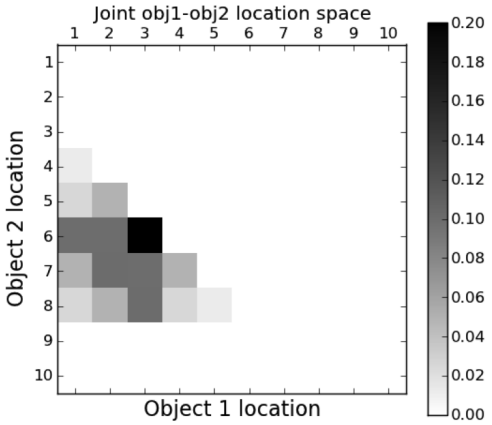
5.4.2 Physical non-interpenetration constraints

When multiple objects are present in the world, a new physical constraint appears: objects cannot interpenetrate each other (Wong et al., 2012). For example, in the 1-D scenario, this means that for any pair of blocks, the one on the right must have location $x^r \geq x^l + L^l$, where x^l and L^l is the location and length of the left block respectively. This is a constraint in the joint state space that couples together all object location/pose variables. One possible solution is to build in the constraint into the domain of $\mathbf{x}^\bullet$ by explicitly disallowing joint states that violate this constraint. Although this is theoretically correct, it forces filtering to be done in the intractable joint space of all object poses, since there is in general no exact way to factor the constraint.

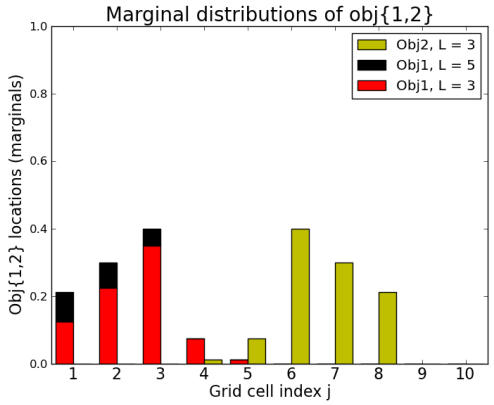
We now consider an alternate way to handle object non-interpenetration that is made possible by considering metric cell occupancies. So far, we have only distinguished between cells being occupied or free, but in the former case there is no indication as to what occupies the cell. In particular, the model so far allows interpenetration because two objects can occupy the same cell, and the cell state being occupied is still consistent. To disallow this, we consider expanding the occupancy attribute for grid cells. I propose splitting the previous 'occupied' value ($\mathbf{m}^j = 1$) into separate values, one for each object index, and one additional value for 'stuff'/unknown. That is, the cell not only indicates that it is occupied, but also which object is occupying it (if known). Then in the object-metric link $\mathbb{P}(\mathbf{m}^j \,|\, \mathbf{x}^\bullet)$, if, for example, obj^2 overlaps $cell^j$, $\mathbf{m}^j$ is enforced to have value 2. The non-interpenetration constraint naturally emerges, since if obj^1 and obj^2 interpenetrate, they must overlap in some cell $cell^j$, whose value is enforced to be both 1 and 2, a situation with zero probability. Such violating joint object states are hence naturally pruned out when evaluating the posterior (Equation 5.14). In particular, even if the object filter's distribution contains violating states with non-zero probability, by considering the objects' effects on grid cells the constraint is enforced and such violating states have zero probability in $\mathbb{P}(\mathbf{x}^\bullet \,|\, \mathbf{z}^\bullet, \mathbf{w}^\bullet)$. We can therefore use a more efficient filter representation that ignores the constraint, such as a product of marginal location



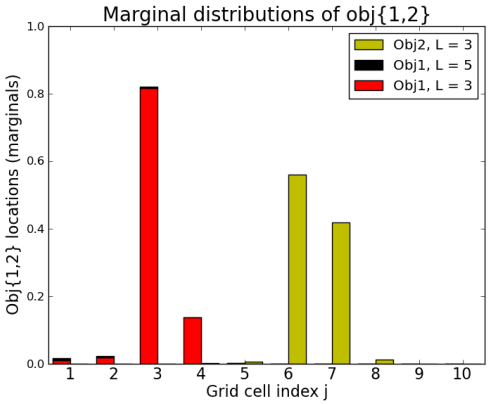
(a) Filter obj^1, obj^2 marginals



(b) Posterior (obj^1, obj^2) joint



(c) Posterior obj^1, obj^2 marginals



(d) Marginals with $cell^2, cell^{10}$ free

Figure 5-5: A 1-D scenario with two objects. When multiple objects are present, a physical non-interpenetration constraint is introduced. (a) The filter maintains the object locations as a product of marginals, which does not respect the constraint. The red/black bars are for obj^1 locations with lengths $L = 3$ and 5 respectively; the yellow bars are for obj^2 locations. (b) After considering the constraint in metric occupancy space, the posterior joint distribution shows that the two object locations are highly coupled. (c) The posterior marginal distributions reflect the constraint's effects: locations in the middle are unlikely either object's left-most cell, because it forces the other object into low-probability states. (d) If additionally $cell^2$ and $cell^{10}$ are observed to likely be free, only a few joint states are possible. Also, the possibility of obj^1 having length $L = 5$ is essentially ruled out. See text in Section 5.4.2 for details.

distributions, and enforce the constraint at query time when metric information is incorporated.

In our 1-D world with two objects, suppose their locations are maintained by the filter as a product of marginal distributions, as depicted in Figure 5-5(a). The marginal distribution for obj^1 is shown in red/black bars; the yellow bars represent the marginal distribution for obj^2. In addition, there is uncertainty in the length of obj^1. Note that this also factors into the non-interpenetration constraint, since, for example, obj^1 at $x^1 = 4$ with $L = 3$ is compatible with obj^2 at $x^2 = 7$, but this is not true for obj^1 with $L = 5$ and the same locations. After enforcing the non-interpenetration constraint by reasoning about metric cell states, the posterior joint object location distribution is shown in Figure 5-5(b). Here white location-pairs have zero probability, and the most likely joint state $(x^1, x^2) = (3, 6)$ has joint probability of 0.2. Based on the marginals and the constraint, obj^1 must be to the left of obj^2, hence the only non-zero probabilities are above the diagonal. The posterior marginal distributions of the two objects' states are depicted in Figure 5-5(c). Locations in the middle are less likely for both objects since, for each object, such locations force the other object into low-probability states. Also, the length L must be 3 the two right-most obj^1 locations, otherwise it will be impossible to fit obj^2 in any locations with non-zero marginal probability.

Suppose we additionally observe that $cell^2$ and $cell^{10}$ are likely to be empty. This greatly restricts the states of the objects; the posterior marginal distributions of the two objects' states in this case is shown in Figure 5-5(d). We see that there are basically only two likely locations for obj^1 now, and that its length is most likely 3 (with probability 0.98). This is because the additional cell observations constrain both objects to be between $cell^2$ and $cell^9$, of which the only object location possibilities are $(x^1, x^2) \in \{(3, 6), (3, 7), (4, 7)\}$. The larger marginal distribution of obj^1 at location 3 is due to the fact that two joint states are possible, and each has relatively high probability from the input marginal distributions given by the objects filter.

In summary, occupancy information can both enforce physical non-interpenetration constraints, as well as reduce uncertainty in object states via free space observations.

5.4.3 Demonstration on robot

I have also empirically validated our approach on a small real-world example, as shown in Figure 5-6 and in an accompanying video (`http://lis.csail.mit.edu/movies/ICRA14_1678_VI_fi.mp4`). The initial setup is shown in Figure 5-6(a): a toy train is placed on a table, and a PR2 robot is attempting to look at it. However, its view is mostly blocked by a board (Figure 5-6(b)); only a small part of the train's front is visible. A simple object instance detector recognizes it as the front of a toy train. The question is, does the train have one car (short) or two cars (long) (Figures 5-6(c) and 5-6(d))? The true answer is one train car in this example.

One way to determine the answer is to move away the occluding board (or equivalently, moving to a better viewpoint). This is depicted by the occupancy grids in Figures 5-6(e)-5-6(g). The grid consists of cubes with side length 2cm, within a 1m $\times$ 0.4m $\times$ 0.2m volume (hence 10^4 cubes in total). The figures show the grid projected onto the table (vertical dimension collapsed). The yellow and black points show free space and occupancy observations respectively. These observations are

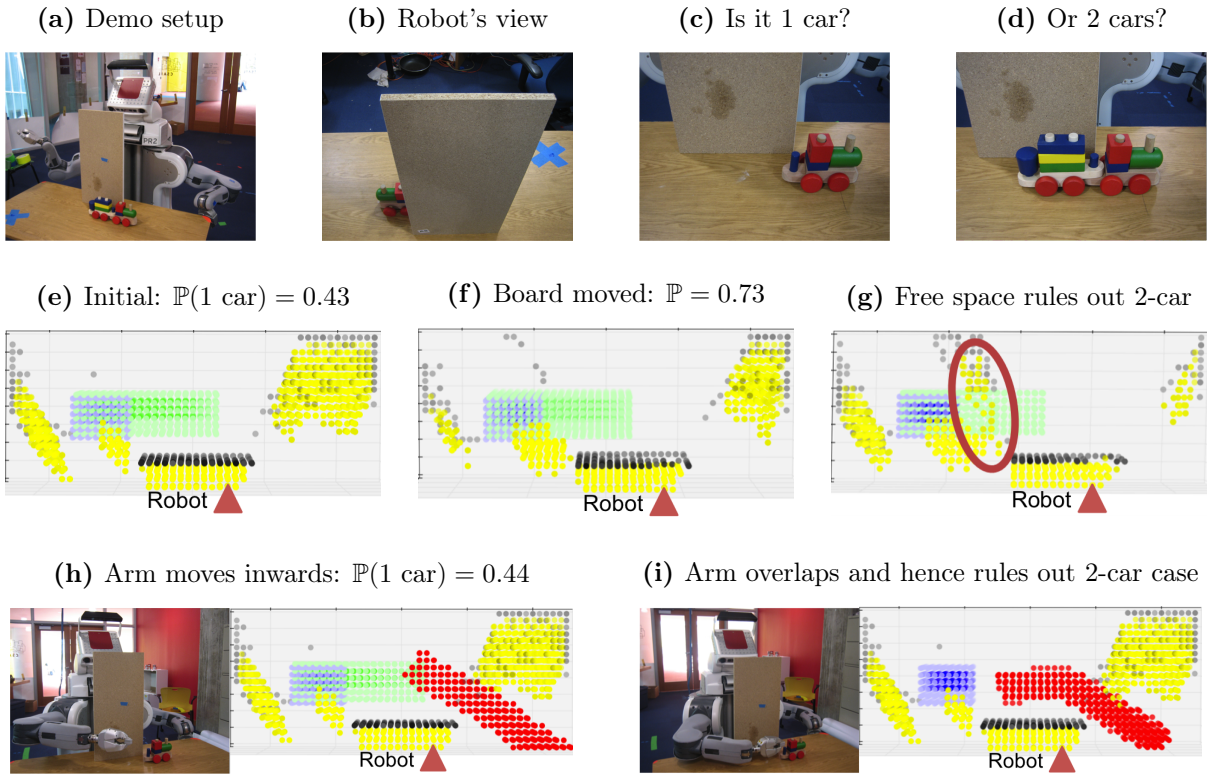


Figure 5-6: A 3-D demonstration on a PR2 robot. Plots show occupancy grids with 1m × 0.4m × 0.2m volume, containing 10^4 cubes of side length 2cm, with the final (vertical) dimension projected onto the table. Colors depict occupancy type/source: Yellow = free space observation; Black = occupancy observation; Blue = inferred occupancy from one-car train; Green = inferred occupancy from two-car train; Red = occupied by robot in its current state. In this projection, the robot is situated at the bottom center of the plot, facing 'upwards'; the black line observed near the bottom corresponds to the board. (a)-(b) A toy train is on a table, but only part of the front is visible to the robot. (c)-(d) This is indicative of two possible scenarios: the train has one car or two cars; there is in fact only one car. (e)-(g) One way to determine the answer is to move the occluding board away. This reveals free space where the second car would have been (circled in (e)), hence ruling out the two-car case. (h)-(i) Another way is to use the robot arm. If the arm successfully sweeps through cells without detecting collision, the cells must have originally been free and are now occupied by the arm. Sweeping through where the second car would have been therefore eliminates the possibility of the train being there. See text in Section 5.4.3 and the accompanying video for details.

determined from depth images returned by a head-mounted Kinect camera: points indicate occupied cells, and rays between points and the camera contain free cells.

Since it is known that there must be a toy train with at least one car, performing object-to-metric inference results in additional cells with *inferred* potential occupancy, as shown by the blue (one car) and green (two car) cases. The number of occupied cells is greater than the train's volume due to uncertainty in the object pose; the cells near the middle have a darker shade because they are more likely to be occupied. As the board is moved gradually to the right, more occupancy observations are collected, and eventually there are free space observations where a second train car should have occupied (circled in Figure 5-6(g)). By inference similar to that from Section 5.4.1, the

two-car case is therefore ruled out.

Without moving either the board or the viewpoint, another way to arrive at the same conclusion is to use the robot arm, shown in Figures 5-6(h) and 5-6(i). Here, occupancy 'observations' (red) are derived from the robot model – cells overlapping the robot in its current configuration must be occupied by the robot. In particular, as in Section 5.4.2, we can augment the occupancy attribute to indicate that these cells are occupied by the robot. As the robot arm sweeps through the space where the second train car would have been, no collisions are detected. This indicates that the space the arm swept through is free or occupied by the robot, which by inference similar to that from Section 5.4.2 rules out the two-car case.

5.5 An Experiment in Comparing Against a Particle Filter

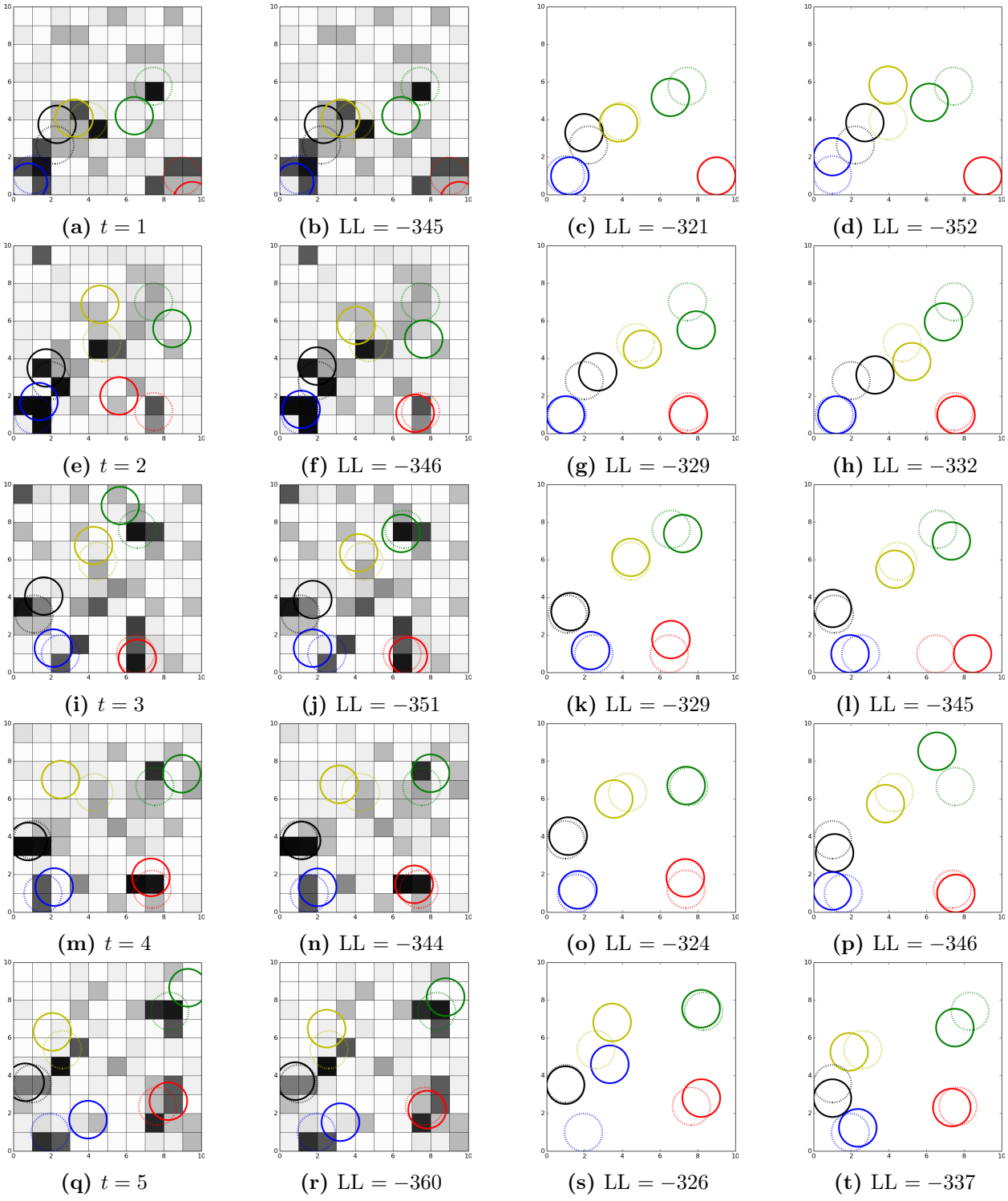
So far, we have discussed potential use cases of the factor-fuse-forget filtering framework presented in this chapter. The filtering strategy was motivated by attempting to avoid filtering in a messy joint/constrained/non-conjugate space. In this section, I present a comparison in simulation between the factored framework and a generic non-parametric particle filter.

The domain consists of 5 objects in a 2-D box of dimensions $[0, 10] \times [0, 10]$. The objects are circular in shape with radius 1, and I will use their centers as the state. For a joint state to be valid, the entirety of each circle must be within the box (i.e., each center's domain is actually $[1, 9] \times [1, 9]$), and no circles should overlap (i.e., pairs of centers should be at least 2 units apart). Over the course of 10 time steps, objects move in a random walk per unit time, each taking a step with a mean-zero isotropic Gaussian distribution, standard deviation 1.0, with the constraint that the resulting joint state must be valid.

At each time, the object is observed once with isotropic Gaussian noise, standard deviation 1.0. The observations may violate the boundary and non-overlap constraints. Objects are completely identifiable, so in this domain there are no identity / data association issues of the sort discussed in the previous chapters. In addition to object observations, there is also an occupancy sensor placed in a grid, with cells being unit squares (i.e., 10×10 grid cells). In each time step, each cell measures its occupancy state 5 times. A cell is considered occupied if and only if its center is overlapped by some object (circle). The occupancy observation is correct with probability 0.7.

Figures 5-7 and 5-8 show the 10 time steps in this domain. The left column of each figure shows the observations given as input to the filters: the solid circles are object measurements, the dashed circles are the true state, and the grid cells are colored by the proportion of occupancy observations that report "occupied" (darker means that the cell received more "occupied" measurements). I compare the factor-and-fuse strategy and a particle filter applied on this data.

In the particle filter, the state is the joint locations of the 5 objects, i.e., the state is 10-dimensional. Starting with 10^4 particles, I iteratively sample valid transitions for each and weigh each by the likelihood of the occupancy observations at each time step. This gives a distribution over joint states; the state with the highest accumulated weight up until the query time is shown



(a) $t = 1$ **(b)** LL $= -345$ **(c)** LL $= -321$ **(d)** LL $= -352$

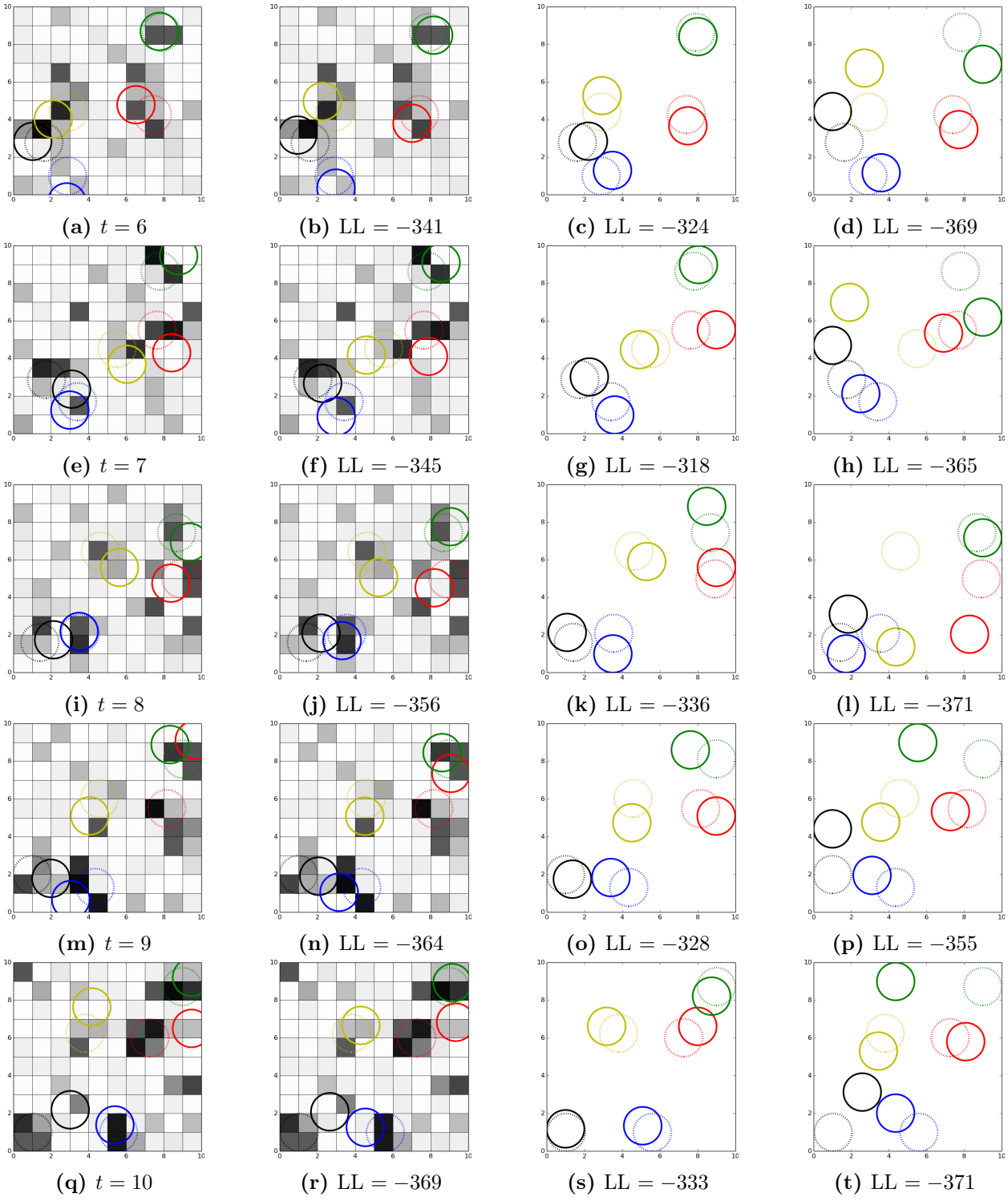
(e) $t = 2$ **(f)** LL $= -346$ **(g)** LL $= -329$ **(h)** LL $= -332$

(i) $t = 3$ **(j)** LL $= -351$ **(k)** LL $= -329$ **(l)** LL $= -345$

(m) $t = 4$ **(n)** LL $= -344$ **(o)** LL $= -324$ **(p)** LL $= -346$

(q) $t = 5$ **(r)** LL $= -360$ **(s)** LL $= -326$ **(t)** LL $= -337$

Figure 5-7: Left column shows observed locations (solid circles) and occupancy (darker = more detections of "occupied"). Objects are color-coded by identity. Dashed circles are true object locations in all frames. Second column shows factored filter, with Gaussian location means (solid circles) and an occupancy grid (darker = more likely occupied). Third and fourth columns are the most-likely samples from fusion (of 10^3) and from a particle filter over object locations (of 10^4). The fused estimate has a significantly greater log-likelihood across all times.



(a) $t = 6$ **(b)** LL $= -341$ **(c)** LL $= -324$ **(d)** LL $= -369$

(e) $t = 7$ **(f)** LL $= -345$ **(g)** LL $= -318$ **(h)** LL $= -365$

(i) $t = 8$ **(j)** LL $= -356$ **(k)** LL $= -336$ **(l)** LL $= -371$

(m) $t = 9$ **(n)** LL $= -364$ **(o)** LL $= -328$ **(p)** LL $= -355$

(q) $t = 10$ **(r)** LL $= -369$ **(s)** LL $= -333$ **(t)** LL $= -371$

Figure 5-8: Left column shows observed locations (solid circles) and occupancy (darker = more detections of "occupied"). Objects are color-coded by identity. Dashed circles are true object locations in all frames. Second column shows factored filter, with Gaussian location means (solid circles) and an occupancy grid (darker = more likely occupied). Third and fourth columns are the most-likely samples from fusion (of 10^3) and from a particle filter over object locations (of 10^4). The fused estimate has a significantly greater log-likelihood across all times.

in the right column of Figures 5-7 and 5-8.

In the factor-and-fuse framework, we first have to supply factored filters. For the object states, I aggressively factor the 5 object states as well, and keep track of each with a 2-D Kalman filter. Since we no longer have joint object states, the non-overlap constraint is not enforced by the factored filters. I also temporarily ignore the boundary constraints. For occupancy data, I use a dynamic occupancy grid to keep track of how likely each of the 10×10 cells is occupied. This is similar to a standard occupancy grid, except it can have Markovian transitions between different occupancy states (Meyer-Delius et al., 2012). In the second columns of Figures 5-7 and 5-8, the Gaussian means of object locations are shown with solid circles, true state with dashed circles, and the occupancy grid by colored cells (darker means that the cell is more likely to be occupied). At certain times (e.g., $t = 8$ and 9), some object mean locations cause objects to overlap / be out of bounds.

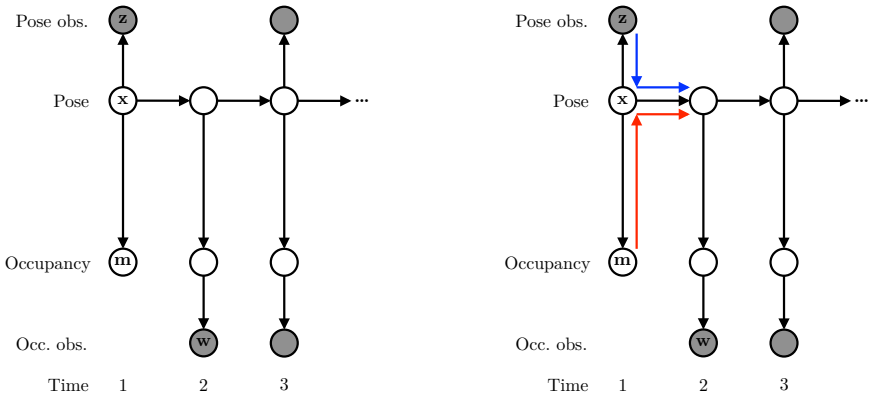
Even prior to fusion, when we compare the log-likelihoods of the factored and particle filter columns, we see that often they are comparable: factored is clearly better at $t \in \{1, 6, 7, 8\}$, similar $t \in \{4, 10\}$, and worse $t \in \{2, 3, 5, 9\}$. Note that factored at this stage has only involved Kalman filters; the occupancy grid has not been used yet.

For fusion, unlike previous examples, we cannot enumerate over the objects' locations and apply Bayes' rule, since the space of locations is continuous in this domain. Instead, I sample each object's location individually from its respective Kalman filter, then concatenate the locations into a joint state. I reject a sample if is violates the boundary constraints (alternatively, I could have sampled from a truncated normal distribution). For the non-interpenetration constraint, I encode that into the occupancy grid as in Section 5.4.2, by giving zero probability to states which involve multiple objects overlapping the same grid cell center. Note that this does not necessarily avoid interpenetration unless the cell width is infinitesimal. A total of 10^3 such concatenated joint states are sampled, and we can assign a weight to each using the occupancy grid and the occupancy prior. The one with the highest such weight is shown in the third-from-left column in Figures 5-7 and 5-8.

As shown both qualitatively and by the log-likelihood scores, the resulting fused estimate is significantly better at all times than the mean/mode of the factored filter, as well as compared to the particle filter, even though the fusion step used 10 times fewer samples. This is not too surprising, because many particles would likely have become degenerate during filtering. Additionally, although we show the fused estimates for all time steps, recall that they do not rely on each other; each fusion step only requires the factored filters at that time. If we were only interested in the joint state at $t = 10$, we can choose not to fuse the previous time steps, and the estimate at $t = 10$ will be the same. This is *not* the case for the particle filter; particles must be propagated through each time step, and hence filtering is much more expensive.

5.6 A Graphical Analysis

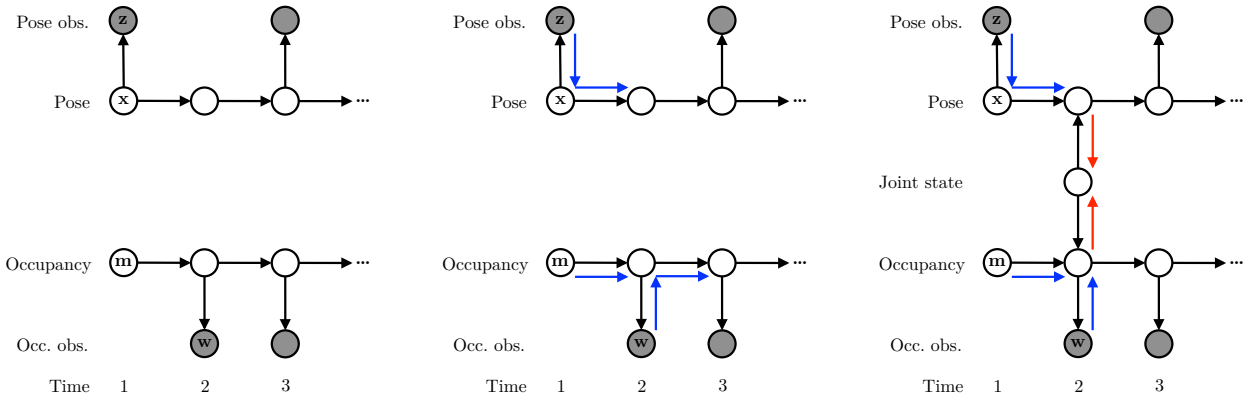
In this section I present additional justification for the presented factor-fuse-forget filtering framework, and consider other related strategies that may be feasible.



(a) Graphical model **(b)** Filtering in pose space only

Figure 5-9: The generic filtering problem within a single representational layer.

The graphical model from Figure 5-1 that illustrates the relationship between the variables of interest in this chapter is replicated in Figure 5-9, unrolled forward in time. As suggested by the graphical model, a natural choice for filtering is to use the pose as the state representation, and attempt to incorporate all observations into the space of poses. The particle filter from the previous section is one such example. On the right figure, we have drawn the information flow as well in colored arrows; blue indicates that data assimilation and state propagation is relatively easy (typically due to conjugacy); red indicates that these steps are potentially difficult. As I argued earlier, incorporating pose observations is easy (e.g., Kalman filter), but incorporating occupancy information generally fragments pose space and requires nonparametric filtering methods.

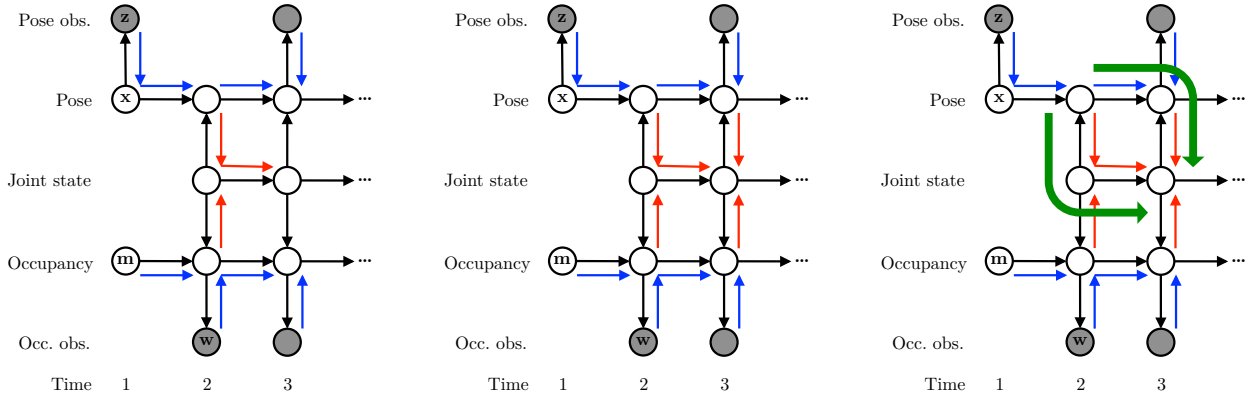


(a) Factored filter: **(b)** Filter independently, **(c)** Fuse on demand

Figure 5-10: The factored filter: Filter independently, fuse on demand.

The strategy I have adopted instead is shown in Figure 5-10. We chose to separate the two state

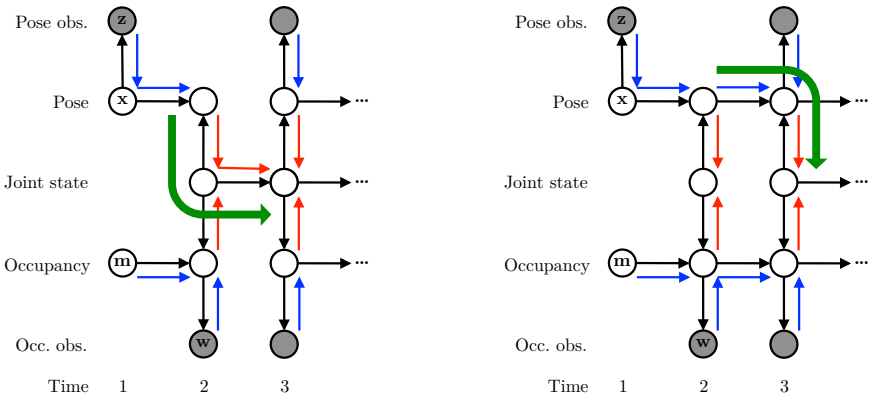
layers, and independently filter in each (left). This requires us to additionally specify a transition function for the occupancy layer. With appropriate choices, filtering in each layer should be relatively efficient (middle). When information from both layers is needed, we perform a typically-expensive fusion step (right).



(a) Propagate joint state? **(b)** When fusing in the future, **(c)** Double-counting will occur

Figure 5-11: If information is propagated forward at too many layers, double-counting will occur.

After fusing, we now have two factored posteriors and a joint posterior distribution. There are multiple ways to proceed with filtering from this point. For example, in the left diagram of Figure 5-11, we can choose to filter on all three layers. This, however, is problematic for two reasons. First, filtering on the fused state is expensive; otherwise, we could have done so from the beginning. Second, information may be double-counted in the future if more fusion steps occur. Suppose, as in the middle diagram, fusion occurs again at the next time step. Then the pose observation z at $t = 1$, which was incorporated into the joint layer via the pose x at $t = 2$, will be fused again via x at $t = 3$. These two paths are indicated by the curved green arrows in the right diagram of Figure 5-11.

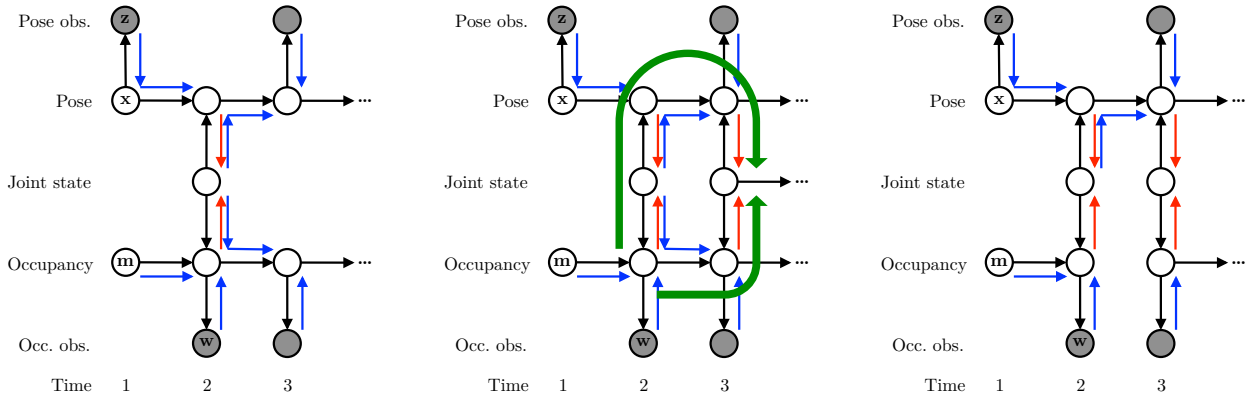


(a) Filter joint state, reset factored filters

(b) Forget joint state, resume factored filters

Figure 5-12: Two viable filtering strategies to avoid double-counting.

The double-counting issue occurs whenever there are multiple paths through which information can travel from one node to another. Hence, to ensure sound filtering, we can keep at most one path. This suggests two possible strategies, shown in Figure 5-12. On the left, we choose to propagate the joint state forward. Then to remove the other arrow, we either have to never fuse again, or we have to reset the factored filters. This solution is not too desirable, since it requires an expensive joint filter, and we also lose the factored filters that may be used more frequently than the joint state. On the right, we choose to propagate the factored states forward but not the joint state. This is exactly our factor-fuse-forget strategy. These two strategies are known for "hierarchical fusion without feedback" fusion architectures, of which our example is an instance (Chong et al., 2012).



(a) Propagate fused information back to factored filters?

(b) If propagate to multiple factors, double-counting may occur

(c) If propagate to one factor, must reset other filters

Figure 5-13: Propagating fused information back to factors may also cause double-counting.

After completing an expensive fusion step, it seems wasteful to simply discard the estimate. Additionally, the joint state may have crucial information that can correct the factored filters; otherwise, the fusion step would be unnecessary. For example, in the previous section, the Kalman filters occasionally lose track of object locations, which the fusion step corrects; it would be useful to be able to project that information back into the assumed Gaussian density (Boyen and Koller, 1998). Consider such a projection scheme in the factor-fuse-forget strategy, as shown in the left diagram of Figure 5-13. The projection step is shown as the blue right arrows emanating from the joint state (typically projection is not too difficult).

However, if we consider a fusion step in the future, as in the center diagram, we see once again that information could have reached the fusion node via multiple paths. In the case shown by the arrows, occupancy observations w at $t = 2$ are fused at $t = 3$ via the occupancy m, and also via the projection from the fusion node at $t = 2$ into the corrected pose x. Once again, we must consider cutting one of the paths. If we ignore the option of disallowing fusion steps, then the only possibilities are to reset either of the factored filters. For example, in the right diagram, if we choose to correct the pose filter, then we must reset the occupancy filter. In general, if we have multiple factored filters, then correcting any single one means that all others must be reset.

5.7 Conclusions and Future Work

Through several examples, I demonstrated that there are many plausible situations in which representing space using both object-based and metric representations is useful and necessary. To combine object-based and metric information, instead of filtering in the complicated joint state space, I adopted a philosophy of filtering in separate, easily-manageable spaces, then only computing fused estimates on demand. To avoid double-counting and computationally-inefficient information propagation, the fused result is forgotten, and filtering resumes in the factored spaces only. This factor-fuse-forget filtering framework for combining object-level and metric-level belief representations was developed extensively in this chapter.

The given examples have been on small, low-dimensional domains. The prospects of directly scaling up the presented approach are unclear. As discussed in Section 5.2.3, the complexity of the generic inference calculation is $O(LX + C)$, where L is the number of cells objects occupy, X is the number of (discrete) attribute settings for all objects, and C is the number of grid cells in the world. Potential efficiencies may be exploited if X is (approximately) factored or if adaptive grids such as octrees are used. Nevertheless, the number of objects and cells needed to represent large spatial environments will still present challenges. Instead, our approach is perhaps most useful for *fine local* estimation: information fusion is *only* performed for few objects/attributes and small areas of great interest (e.g., to a given task), in cases where information from either the object-level or metric-level representation alone is insufficient.

More theoretical and empirical work is needed to determine the ramifications of our representation when used in large environments over long periods of time. It would be interesting to find out what approximations the strategy is making exactly, which would inform us about cases in which it would likely work or fail. Handling continuous and high-dimensional state (attribute) spaces, as well as scaling up to larger environments containing many objects, are subjects of future work. Nevertheless, even in its current simplistic and generic form, our approach enables novel lines of spatial inference that could not be accomplished using single layers of spatial representation.

Chapter 6

Conclusion

In this dissertation, I have argued that object-based world models are needed for mobile-manipulation robots operating in typical indoor environments, and that existing spatial representations such as feature-based SLAM and semantic maps will not be sufficient for mobile manipulators that wish to reason and act intelligently in the world. I presented three aspects of the world modeling problem, where the common technical thread was that information had to be aggregated in some manner; respectively, across space, time, and sensor modalities. The key challenge in accomplishing this fusion is a data association problem, and a potential mismatch between different state and belief representations. The key contributions were a new clustering-based batch data association paradigm using Bayesian nonparametric models, and an unconventional factor-fuse-forget filtering framework.

There are many obvious extensions to the described work. Since generating more questions than answers is a hallmark of good research questions, hopefully, while being wary of the fallacy of confusing necessity and sufficiency, this indicates the possibility that object-based world modeling is an interesting topic with further research potential. Approach-specific extensions, such as leveraging fast algorithms for performing inference in DPMMs, were discussed in the discussion sections of previous chapters. I will close our journey in this dissertation with brief thoughts on two questions: What can we do with the object-based world model now? How might we go further?

6.1 What can we do now?

One major inspiration for my dissertation work was the object search work I did early in my graduate career Wong et al. (2013). There, I 'built character' in Bayesian methods by gradually constructing a custom graphical model for inferring potential locations of unseen objects, using object-object co-occurrence information and capacity constraints (see Figure 6-1). This model was created based on two intuitions: objects of similar types tend to be co-located, and large unseen objects cannot fit in small unexplored regions. To come full circle, I believe this same model and inference procedure can naturally come out of our current general world model. In particular, using our world model, we should be able to track the poses of many objects over time, and from this data infer the likely locations and possibly co-occurrence statistics of object pairs. Then, together with an occupancy

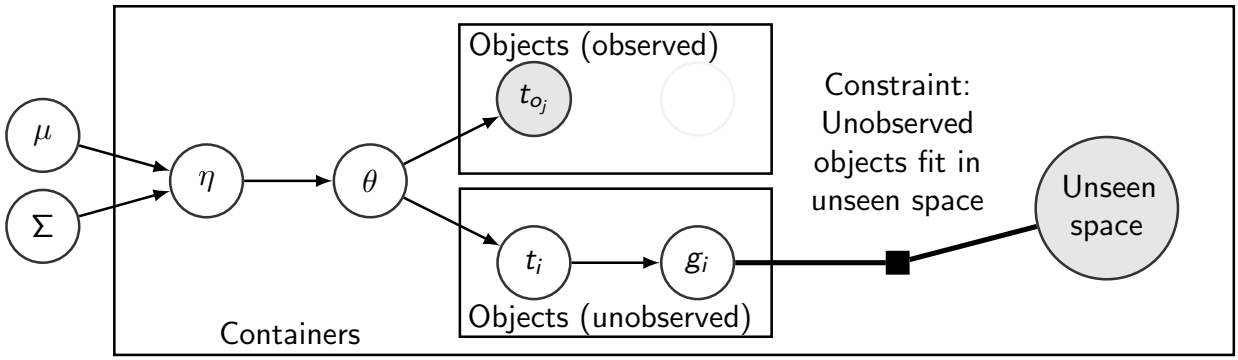


Figure 6-1: A probabilistic model for inferring locations of unseen objects.

grid representing our current explored regions, we should be able to fuse this information with object pose predictions to automatically deduce capacity constraints. These inferences can then guide the robot to take useful manipulation actions, possibly even predicting what might appear after an object is moved away and unexplored space is revealed.

More generally, the world model estimates the state of the world and tracks our uncertainty (with respect to object attributes and occupancy). Since this is a type of belief about the world, we should be able to use it to guide actions; connecting to an action selection strategy / planning algorithm is clearly a next step. For example, we can perform the usual next-best-view information gathering actions. Another interesting direction is to use the world model to track our state of *ignorance* in the world's objects – for example, many occupied cells in an occupancy grid may not be overlapping any known objects in the world model. This may indicate that a collection of unidentified cells may correspond to an unknown object type, and the robot can then attempt to *learn* about novel objects. For any world model to be practical in human environments, I envision that it must have the capability to adapt and grow its representation.

6.2 How might we go further?

The point above raises an interesting counterpoint: world models, and estimators in general, cannot *just* grow in size. Even with the proof-of-concept domains shown in this dissertation, inference was already non-trivial. Consider the scenario shown in Figure 6-2: there are many more objects, additional attributes of interest, large regions of space, compounded over long time horizons. Attempting to directly scale up our current model to a problem of this size is hopeless. Instead, I argue that world models and estimators must also have the capability to *compress* its representation, by aggressively 'forgetting' information, or simply ignoring it.

Aggressively forgetting is in the same vein as heavy pruning in methods such as the MHT. However, we also know that simply discarding information permanently without maintaining a sufficient statistic eventually leads to errors. Since computational tractability demands a pruning strategy, perhaps what we need instead is a *recovery* mechanism, triggered by a *fault diagnosis/identification* process. For example, for online purposes, we filter incoming information using an aggressively-

Figure 6-2: A Willow Garage PR2 robot in a typical laboratory environment.

pruned MHT, but also store a significantly longer historical snapshot. If, through some inference procedure or failed-assertions triggers, a *meta-estimator* comes to believe that the MHT has diverged from the true state, then the stored history is consulted and used to reset the filter, possibly by using more robust but slower solutions such as clustering-based batch data association.

Perhaps even harder, but certainly necessary eventually, is allowing estimators to learn to ignore certain spaces of information. This is based on the recognition that for any given task, most of the world state is typically irrelevant. For example, I certainly function properly in my office without worrying about the state of things in my home. I currently fundamentally believe that:

> Estimators, including world models, must be tied to the task.

In the long run, since the task changes, what we need is a method for agents to automatically construct, adapt, and reconfigure estimators that can be used in different tasks. Choice of state representation and estimator, conventionally a privilege and responsibility that is solely granted to human system designers, should be made available to all intelligent agents, natural or artificial.

6.3 Final Remarks

Instead of contributing to the sea of ending remarks, I close this dissertation by quoting an observation made by Anderson and Oates (2007) that succinctly summarizes the current state of affairs:

> Natural intelligent systems tend to be robust;
> Artificial intelligent systems tend to be brittle.

Bibliography

M. Adams, B.-N. Vo, R. Mahler, and J. Mullane. SLAM gets a PHD: New concepts in map estimation. *IEEE Robotics Automation Magazine*, 21(2):26–37, 2014.

A. Ahmed and E. Xing. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering. In *SIAM International Conference on Data Mining*, 2008.

A. Aldoma, F. Tombari, J. Prankl, A. Richtsfeld, L. Di Stefano, and M. Vincze. Multimodal cue integration through hypotheses verification for RGB-D object recognition and 6DOF pose estimation. In *IEEE International Conference on on Robotics and Automation*, 2013.

D.J. Aldous. Exchangeability and related topics. In P.L. Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XIII*, pages 1–198. Springer Berlin Heidelberg, 1985.

R. Anati, D. Scaramuzza, K.G. Derpanis, and K. Daniilidis. Robot localization using soft object detection. In *IEEE International Conference on Robotics and Automation*, 2012.

M.L. Anderson and T. Oates. A review of recent research in metareasoning and metalearning. *AI Magazine*, 28(1):7–16, 2007.

C.E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152âĂŞ–1174, 1974.

N. Atanasov, B. Sankaran, J. Le Ny, T. Koletschka, G. Pappas, and K. Daniilidis. Hypothesis testing framework for active object detection. In *International Conference on Robotics and Automation*, 2013.

N. Atanasov, M. Zhu, K. Daniilidis, and G.J. Pappas. Localization from semantic observations via the matrix permanent. *The International Journal of Robotics Research*, 35(1-3):73–99, 2016.

Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

Y. Bar-Shalom, P.K. Willett, and X. Tian. *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011.

T. Bayes and R. Price. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions*, 53:370–418, 1763.

J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.

J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer=Verlag New York, 2006.

D. Blackwell and J.B. MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

D.M. Blei and M.I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

G.E.P. Box and G.C. Tiao. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons, 1973.

X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Uncertainty in Artificial Intelligence*, 1998.

W. Burgard and M. Hebert. World modeling. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 853–869. Springer-Verlag Berlin Heidelberg, 2008.

T. Campbell, M. Liu, B. Kulis, J.P. How, and L. Carin. Dynamic clustering via asymptotics of the dependent Dirichlet process mixture. In *Advances in Neural Information Processing Systems*, 2013.

C.-Y. Chong, K.-C. Chang, and S. Mori. Fundamentals of distributed estimation. In D. Hall, C.-Y. Chong, J. Llinas, and M. Liggins II, editors, *Distributed Data Fusion for Network-Centric Operations*, pages 95–124. CRC Press, 2012.

I.J. Cox and S.L. Hingorani. An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

I.J. Cox and J.J. Leonard. Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, 1994.

J.L. Crowley. Dynamic world modeling for an intelligent mobile robot using a rotating ultra-sonic ranging device. In *IEEE International Conference on Robotics and Automation*, 1985.

D.B. Dahl. Sequentially-allocated merge-split sampler for conjugate and nonconjugate Dirichlet process mixture models. Technical report, Texas A&M University, 2005.

D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer-Verlag New York, 1988.

P. Dames and V. Kumar. Autonomous localization of an unknown number of targets without data association using teams of mobile sensors. *IEEE Transactions on Automation Science and Engineering*, 12(3):850–864, 2015.

P. Dames, D. Thakur, M. Schwager, and V. Kumar. Playing fetch with your robot: The ability of robots to locate and interact with objects. *IEEE Robotics Automation Magazine*, 21(2):46–52, 2014.

H. Daumé III. Fast search for Dirichlet process mixture models. In *International Conference on Artificial Intelligence and Statistics*, 2007.

F. Dellaert. *Monte Carlo EM for Data-Association and its Applications in Computer Vision*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2001.

F. Dellaert, S.M. Seitz, C.E. Thorpe, and S. Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50(1–2):45–71, 2003.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag New York, 2001.

H. Durrant-Whyte and T.C. Henderson. Multisensor data fusion. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 585–610. Springer-Verlag Berlin Heidelberg, 2008.

R. Eidenberger and J. Scharinger. Active perception and scene modeling by planning with probabilistic 6D object poses. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

S. Ekvall, D. Kragic, and P. Jensfelt. Object detection and mapping for service robot tasks. *Robotica*, 25(2):175–187, 2007.

J. Elfring, S. van den Dries, M.J.G. van de Molengraft, and M. Steinbuch. Semantic world modeling using probabilistic multiple hypothesis anchoring. *Robotics and Autonomous Systems*, 61(2):95–105, 2013.

M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1 (2):209–230, 1973.

C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigál, and J. González. Multi-hierarchical semantic maps for mobile robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian Data Analysis*. CRC Press, 3rd edition, 2013.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

S.J. Gershman and D.M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

Z. Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2013.

J.K. Ghosh and R.V. Ramamoorthi. *Bayesian Nonparametrics*. Springer-Verlag New York, 2003.

J. Glover and S. Popovic. Bingham Procrustean alignment for object detection in clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

J. Glover, R.B. Rusu, and G. Bradski. Monte Carlo pose estimation with quaternion kernels and the Bingham distribution. In *Robotics: Science and Systems*, 2011.

I.R. Goodman, R.P.S. Mahler, and H.T. Nguyen. *Mathematics of Data Fusion*. Springer Netherlands, 1997.

G.D. Hager and B. Wegbreit. Scene parsing using a prior world model. *The International Journal of Robotics Research*, 30(12):1477–1507, 2011.

W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

E. Herbst and D. Fox. Mapping as a service. In *AAAI Conference on Artificial Intelligence Workshop on Intelligent Robotic Systems*, 2013.

N.L. Hjort, C. Holmes, P. Müller, and S.G. Walker, editors. *Bayesian Nonparametrics*. Cambridge University Press, 2010.

R. Huang, F. Zhu, and P.-A. Heng. The dynamic Chinese restaurant process via birth and death processes. In *AAAI Conference on Artificial Intelligence*, 2015.

A.K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

S. Jain and R.M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004.

L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

B. Kalyan, K.W. Lee, and W.S. Wijesoma. FISST-SLAM: Finite set statistical approach to simultaneous localization and mapping. *The International Journal of Robotics Research*, 29(10): 1251–1262, 2010.

R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller, J.W. Thatcher, and J.D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Springer US, 1972.

B. Khaleghi, A. Khamis, F.O. Karray, and S.N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

A.N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea Publishing Company, 2nd edition, 1956. English translation.

K. Konolige, E. Marder-Eppstein, and B. Marthi. Navigation in hybrid metric-topological maps. In *IEEE International Conference on Robotics and Automation*, 2011.

I. Kostavelis and A. Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86–103, 2015.

M.C. Koval, N.S. Pollard, and S.S. Srinivasa. Pose estimation for planar contact manipulation with manifold particle filters. *The International Journal of Robotics Research*, 34(7):922–945, 2015.

H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955.

B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119(1–2):191–233, 2000.

B. Kulis and M.I. Jordan. Revisiting k-means: New algorithms via Bayesian nonparametrics. In *International Conference on Machine Learning*, 2012.

T. Kurien. Issues in the design of practical multitarget tracking algorithms. In Y. Bar-Shalom, editor, *Multitarget-Multisensor Tracking: Advanced Applications*, pages 43–84. Artech House, 1990.

K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3D scenes. In *IEEE International Conference on on Robotics and Automation*, 2012.

M.E. Liggins, D.L. Hall, and J. Llinas, editors. *Handbook of Multisensor Data Fusion: Theory and Practice*. CRC Press, 2nd edition, 2008.

D. Lin. *Generative Modeling of Dynamic Visual Scenes*. PhD thesis, Massachusetts Institute of Technology, 2012.

D. Lin, E. Grimson, and J. Fisher. Construction of dependent Dirichlet processes based on Poisson processes. In *Advances in Neural Information Processing Systems*, 2010.

Z. Liu and G. von Wichert. Extracting semantic indoor maps from occupancy grids. *Robotics and Autonomous Systems*, 62(5):663–674, 2014.

W. Luo, J. Xing, X. Zhang, X. Zhao, and T.-K. Kim. Multiple object tracking: A literature review. *ArXiv e-prints*, 2014.

W. Luo, B. Stenger, X. Zhao, and T.-K. Kim. Automatic topic discovery for multi-object tracking. In *AAAI Conference on Artificial Intelligence*, 2015.

S.N. MacEachern. Dependent nonparametric processes. In *ASA Section on Bayesian Statistics*, 1999.

S.N. MacEachern. Dependent Dirichlet processes. Technical report, Ohio State University, 2000.

R. Mahler. "Statistics 102" for multisource-multitarget detection and tracking. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):376–389, 2013.

R.P.S. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.

R.P.S. Mahler. "Statistics 101" for multisensor, multitarget data fusion. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):53–64, 2004.

R.P.S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007.

R.P.S. Mahler. *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, 2014.

Z.-C. Marton, F. Balint-Benczedi, O.M. Mozos, N. Blodow, A. Kanezaki, L.C. Goron, D. Pangercic, and M. Beetz. Part-based geometric categorization and object reconstruction in cluttered table-top scenes. *Journal of Intelligent & Robotic Systems*, pages 1–22, 2014.

J. Mason and B. Marthi. An object-based semantic world model for long-term change detection and semantic querying. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

S.B. McGrayne. *The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy*. Yale University Press, 2011.

N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

D. Meyer-Delius, M. Beinhofer, and W. Burgard. Occupancy grid models for robot mapping in changing environments. In *AAAI Conference on Artificial Intelligence*, 2012.

J.W. Miller. *Nonparametric and Variable-Dimension Bayesian Mixture Models: Analysis, Comparison, and New Methods*. PhD thesis, Brown University, 2014.

J.W. Miller and M.T. Harrison. A simple example of Dirichlet process mixture inconsistency for the number of components. In *Advances in Neural Information Processing Systems*, 2013.

J.W. Miller and M.T. Harrison. Inconsistency of Pitman-Yor process mixtures for the number of components. *Journal of Machine Learning Research*, 15:3333–3370, 2014.

J.W. Miller and M.T. Harrison. Mixture models with a prior on the number of components. *ArXiv e-prints*, 2015.

H. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation*, 1985.

J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo. *Random Finite Sets for Robot Mapping and SLAM: New Concepts in Autonomous Robotic Map Representations*. Springer Berlin Heidelberg, 2011.

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

K.G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3):682–687, 1968.

R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

W. Neiswanger, F. Wood, and E.P. Xing. The dependent Dirichlet process mixture of objects for detection-free tracking and object modeling. In *International Conference on Artificial Intelligence and Statistics*, 2014.

R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, 2011.

A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

S. Oh, S. Russell, and S. Sastry. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Transactions on Automatic Control*, 54(3):481–497, 2009.

P. Orbanz and Y.W. Teh. Bayesian nonparametric models. In C. Sammut and G.I. Webb, editors, *Encyclopedia of Machine Learning*, pages 81–89. Springer US, 2010.

J.L. Pacheco and E.B. Sudderth. Improved variational inference for tracking in clutter. In *IEEE Statistical Signal Processing Workshop (SSP)*, 2012.

H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *International Joint Conference on Artificial Intelligence*, 1999.

A. Pronobis and P. Jensfelt. Large-scale semantic mapping and reasoning with heterogeneous modalities. In *IEEE International Conference on Robotics and Automation*, 2012.

A. Pronobis, O.M. Mozos, B. Caputo, and P. Jensfelt. Multi-modal semantic place classification. *The International Journal of Robotics Research*, 29(2–3):298–320, 2010.

A. Ranganathan and F. Dellaert. Semantic modeling of places using objects. In *Robotics: Science and Systems*, 2007.

H.E. Rauch, F. Tung, and C.T. Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

Y.P. Raykov, A. Boukouvalas, and M.A. Little. Simple approximate MAP inference for Dirichlet processes. *ArXiv e-prints*, 2014.

D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer New York, 2nd edition, 2004.

C.P. Robert. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer-Verlag New York, 2007.

R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3D recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

L.D. Stone, R.L. Streit, T.L. Corwin, and K.L. Bell. *Bayesian Multiple Target Tracking*. Artech House, 2nd edition, 2014.

E.B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

Y.W. Teh. Dirichlet processes. In C. Sammut and G.I. Webb, editors, *Encyclopedia of Machine Learning*, pages 280–287. Springer US, 2010.

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

S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1âĂŞ2):99–141, 2001.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots – an object based approach. *Robotics and Autonomous Systems*, 55(5):359–371, 2007.

J. Velez, G. Hemann, A.S. Huang, I. Posner, and N. Roy. Modelling observation correlations for active exploration and robust object detection. *Journal of Artificial Intelligence Research*, 44: 423–453, 2012.

B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, 2006.

B.-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245, 2005.

L. Wang and D.B. Dunson. Fast Bayesian inference in Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 20(1):196–216, 2011.

L. Wasserman. *All of Nonparametric Statistics*. Springer New York, 2006.

T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J.J. Leonard, and J. McDonald. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *The International Journal of Robotics Research*, 34(4–5):598–626, 2015.

L.L.S. Wong, L.P. Kaelbling, and T. Lozano-Pérez. Collision-free state estimation. In *IEEE International Conference on Robotics and Automation*, 2012.

L.L.S. Wong, L.P. Kaelbling, and T. Lozano-Pérez. Manipulation-based active search for occluded objects. In *IEEE International Conference on Robotics and Automation*, 2013.

L.L.S. Wong, L.P. Kaelbling, and T. Lozano-Pérez. Not seeing is also believing: Combining object and metric spatial information. In *IEEE International Conference on Robotics and Automation*, 2014.

L.L.S. Wong, L.P. Kaelbling, and T. Lozano-Pérez. Data association for semantic world modeling from partial views. *The International Journal of Robotics Research*, 34(7):1064–1082, 2015.

K.M. Wurm, D. Hennes, D. Holz, R.B. Rusu, C. Stachniss, K. Konolige, and W. Burgard. Hierarchies of octrees for efficient 3D mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

K. Wyffels and M. Campbell. Negative information for occlusion reasoning in dynamic extended multiobject tracking. *IEEE Transactions on Robotics*, 31(2):425–442, 2015.

H. Zender, O.M. Mozos, P. Jensfelt, G.-J.M. Kruijff, and W. Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.

X. Zhu, Z. Ghahramani, and J. Lafferty. Time-sensitive Dirichlet process mixture models. Technical report, Carnegie Mellon University, 2005.