ACTIVE INFORMATION ACQUISITION WITH MOBILE ROBOTS

Nikolay Asenov Atanasov

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

2015

George J. Pappas, Professor of Electrical and Systems Engineering
Supervisor of Dissertation

Kostas Daniilidis, Professor of Computer and Information Science
Co-Supervisor of Dissertation

Alejandro Ribeiro, Associate Professor of Electrical and Systems Engineering
Graduate Group Chairperson

Dissertation Committee:
Daniel D. Lee, Professor of Electrical and Systems Engineering
Vijay Kumar, Professor of Mechanical Engineering and Applied Mechanics
Jerome Le Ny, Assistant Professor of Electrical Engineering, Polytechnique Montreal
Stefano Soatto, Professor of Computer Science, University of California Los Angeles

ACTIVE INFORMATION ACQUISITION WITH MOBILE ROBOTS

To my family.

Acknowledgments

This thesis would not have been possible without my advisors, George Pappas, Kostas Daniilidis, and Jerome Le Ny. My deepest thanks go to George for choosing me as his graduate student, teaching me how to think independently, and helping me remember what the point of this huge endeavor is. Thanks to George I learned to communicate my ideas and excite my peers about them. His guidance and support have been my driving force through the years and have shaped my academic identity.

I also thank Kostas for being a source of inspiration, always positive and excited about our research. Our discussions always led to amazing ideas for research problems and interesting applications. Most importantly, Kostas motivated me and taught me to enjoy and respect the research process. I am extremely grateful to Jerome for mentoring me and providing support in the difficult moments. Together, we discussed at length both successful and unsuccessful ideas, proofs, and algorithms.

I thank my dissertation committee, Stefano Soatto, Dan Lee, and Vijay Kumar, for putting up with scheduling constraints, wanting to learn about my ideas, and reviewing my work. I thank my co-authors, Bharath Sankaran, Menglong Zhu, Nathan Michael, and Roberto Tron, for helping with a lot of the results and experiments presented in this thesis. I thank Ben Charrow and Phil Dames for our fruitful discussions on information gathering. I thank Hadi, Shahin, Konstantinos, and Omur for the wonderful times we spent doing homework, discussing the challenges of graduate school, and pondering about the future. I had amazing support from my closest friends Petko, Kalin, Rado, Vasil, Yavor, and Kiro. I also thank my undergraduate school advisors Taikang Ning and David Ahlgren for introducing me to engineering research and inspiring me to purse graduate education.

Finally, I thank my family, Asen, Snezhanka, and Vili, for supporting me, listening to my complaints, sharing my happiness, always trying to understand me, and providing advice. I also thank my grandparents Dafina and Nikolay Atanasov and Velika and Ivan Uzunov for being curious about my work and discussing my plans and aspirations. Most importantly, I thank my love, Huanhuan, for her understanding and support during my research career.

ABSTRACT

ACTIVE INFORMATION ACQUISITION WITH MOBILE ROBOTS

Nikolay A. Atanasov
George J. Pappas
Kostas Daniilidis

The recent proliferation of sensors and robots has potential to transform fields as diverse as environmental monitoring, security and surveillance, localization and mapping, and structure inspection. One of the great technical challenges in these scenarios is to control the sensors and robots in order to extract accurate information about various physical phenomena *autonomously*. The goal of this dissertation is to provide a unified approach for active information acquisition with a team of sensing robots. We formulate a decision problem for maximizing relevant information measures, constrained by the motion capabilities and sensing modalities of the robots, and focus on the design of a scalable control strategy for the robot team.

The first part of the dissertation studies the active information acquisition problem in the special case of linear Gaussian sensing and mobility models. We show that the classical principle of separation between estimation and control holds in this case. It enables us to reduce the original stochastic optimal control problem to a deterministic version and to provide an optimal centralized solution. Unfortunately, the complexity of obtaining the optimal solution scales exponentially with the length of the planning horizon and the number of robots. We develop approximation algorithms to manage the complexity in both of these factors and provide theoretical performance guarantees. Applications in gas concentration mapping, joint localization and vehicle tracking in sensor networks, and active multi-robot localization and mapping are presented. Coupled with linearization and model predictive control, our algorithms can even generate adaptive control policies for nonlinear sensing and mobility models.

Linear Gaussian information seeking, however, cannot be applied directly in the presence of sensing nuisances such as missed detections, false alarms, and ambiguous data association or when some sensor observations are discrete (e.g., object classes, medical alarms) or, even worse, when the sensing and target models are entirely unknown. The second part of the dissertation considers these complications in the context of two applications: active localization from semantic observations (e.g, recognized objects) and radio signal source seeking. The complexity of the target inference problem forces us to resort to greedy planning of the sensor trajectories.

Non-greedy closed-loop information acquisition with general discrete models is achieved in the final part of the dissertation via dynamic programming and Monte Carlo tree search algorithms. Applications in active object recognition and pose estimation are presented. The techniques developed in this thesis offer an effective and scalable approach for controlled information acquisition with multiple sensing robots and have broad applications to environmental monitoring, search and rescue, security and surveillance, localization and mapping, precision agriculture, and structure inspection.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The heart of data-enabled science is the extraction and interpretation of information residing in dynamic hard-to-predict phenomena. Classically, information is collected by static sensors and is summarized in spatiotemporal models, represented in the language of Bayesian filters and graphical models. The stance of this thesis is that going beyond static data collection will enable intelligent sampling and will produce higher-fidelity models efficiently. Autonomous sensor deployments are also adaptable to environment conditions and will reduce human supervision and allow access to dangerous areas. The effective coordination of mobile sensing resources relies on and aims at improving the models produced at the inference level. Due to this coupling, information-gathering problems are quite challenging and necessitate the development of new principled approaches for dealing with the underlying complexity. A small tweak in the existing body of work is unlikely to yield the desired results. In particular, there is a need for breakthroughs in hierarchical representations, distributed inference, and adaptive planning algorithms, which are scalable both in the length of the planning horizon and in the number of sensors and can handle heterogeneity both in the environment models and in the vehicle dynamics. In exchange, the rewards are compelling. Distributing the estimation and control will allow robust coordination of numerous intelligent systems. Heterogeneity will enable collaboration among water, ground, and air resources with various sensing capabilities, thereby dramatically improving the awareness of the sensor team at metric, topological, and semantic levels. Such developments will have a tremendous impact in the following fields:

- Environmental monitoring (Fig. 1.1): air, water, and soil quality, temperature, pressure, magnetic forces, chemical concentration, diffusivity, bio-diversity
- Agriculture
- Construction and structure inspection (Fig. 1.2)
- Security and surveillance (Fig. 1.3 & Fig. 1.4)
- Search and rescue operations
- Object detection, classification, and pose estimation (Fig. 1.6)

- Localization and mapping (Fig. 1.5)
- Mining
- Space exploration

Therefore, the focus of this dissertation is on active information acquisition with mobile robots and configurable sensing systems. In the next section, we capture the characteristics of the aforementioned scenarios in a precise problem formulation. The objective is to design a scalable control strategy for the robot team in order to tracking evolving phenomena of interest accurately and efficiently.

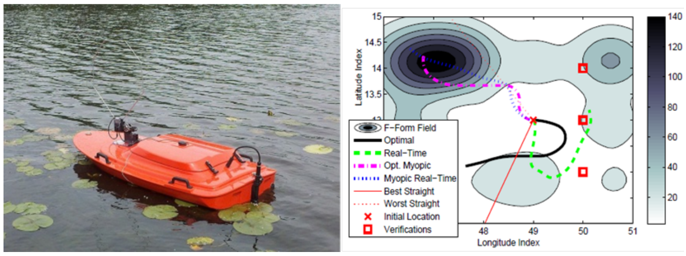


Figure 1.1: Autonomous water vehicle tracking radio-emitter-tagged invasive carp species in Minnesota (courtesy of Tokekar et al. 2013 and Choi 2009).

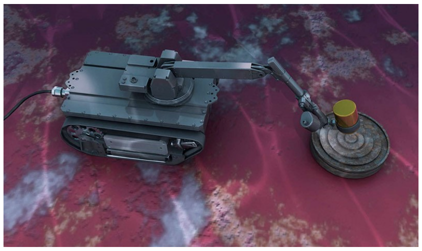


Figure 1.2: Autonomous inspection of the hull integrity of a submerged ship.

Figure 1.3: A military quadrotor patrolling an area of interest.

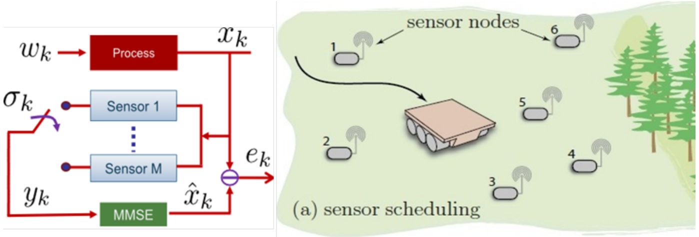


Figure 1.4: Scheduling communication power, sensor use, and operational parameters in sensor networks is crucial for accurate and efficient tracking of physical phenomena (courtesy of Huber 2009).

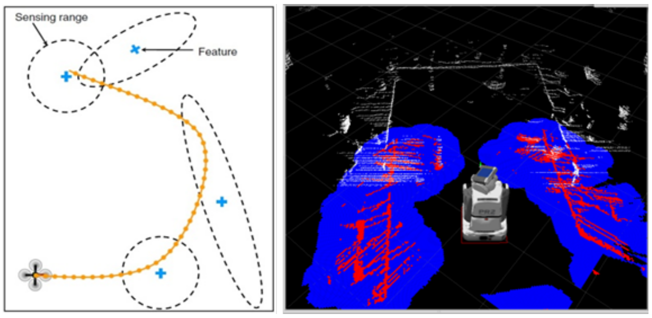


Figure 1.5: A robot can improve its localization and environment map by planning an informative trajectory for its future observations (courtesy of Vitus et al. 2012)

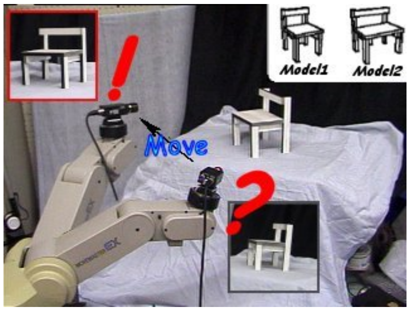


Figure 1.6: Planning the viewpoint of an autonomous camera in order to improve the results of object recognition

1.2 The Active Information Acquisition Problem

Information acquisition is the process of estimating the state of an observed system of interest by utilizing the available sensing modalities. Consequently, approaches from estimation theory, such as filtering, smoothing, and belief propagation, form the basis of information acquisition. Active information acquisition, on the other hand, is the process of planning future trajectories and configurations for the sensing systems, over multiple time steps, in order to improve the performance of the estimation process. It couples the estimation problem with a stochastic control problem, which requires making decisions based on anticipated, not-yet-known sensor measurements.

The starting point of an active information gathering algorithm is a collection of three dynamical models which capture mathematically the degrees of freedom of our sensors, the evolution of the physical process, and relationship between the sensor observations and the target process. Having designed these models, one can proceed with the construction of an estimator which uses the sensor observations to refine the knowledge about the state of the underlying physical process. The final step is to plan the motion and configurations of the sensors in order to collect maximally-informative observations, thus optimizing the estimation process. This requires a proper choice of a reward function to judge the informativeness of the planned sensor trajectories. Fig. 1.7 presents an overview of the active information acquisition process.

Proceeding more formally, consider a team of n mobile sensors with states $x_{i,t},\ i = 1,\dots,n$ at time t and dynamics governed by the following discrete-time continuous-state *sensor motion model*:

$$x_{i,t+1} = f_i(x_{i,t}, u_{i,t}, \text{noise}), \tag{1.1}$$

where $u_{i,t}$ is the control input. To simplify notation, denote the collection of all sensor states by $x_t := [x_{1,t}^T \cdots x_{n,t}^T]^T$, the collection of all inputs by $u_t := [u_{1,t}^T \cdots u_{n,t}^T]^T$, and the combined sensor motion model by $x_{t+1} = f(x_t, u_t, \text{noise})$. In many applications, the sensor mobility is due to the fact that the sensors are mounted on a vehicle or a robot. For example, the kinematics of a ground robot with nonholonomic (car-like) constraints can be described by a differential-drive model (Appendix B.1).

The task of the sensors is to track the evolution of a collection of targets (e.g., mobile vehicles, visual landmarks, radio transmitters, gas concentration in different regions of the environment, etc.) with joint state y_t. The target evolution is governed by a *target motion model*:

$$y_{t+1} = a(y_t, \text{noise}). \tag{1.2}$$

For example, the behavior of a mobile target in tracking scenarios is often described by a constant-velocity motion model (Appendix B.2).

Each sensor receives measurements $z_{i,t}$ of the targets, governed by the following *sensor observation model*:

$$z_{i,t} = h_i(x_{i,t}, y_t, \text{noise}). \tag{1.3}$$

Again, to simplify notation, we use $z_t := [z_{1,t}^T \cdots z_{n,t}^T]^T$ and denote the combined observation model by $z_t = h(x_t, y_t, \text{noise})$. Some widely-used sensing models are provided in Appendix B. For example, a laser scanner can be modeled as a range sensor (Appendix B.3), a directional antenna - as a bearing sensor (Appendix B.4), a stereo camera - as a stereo sensor

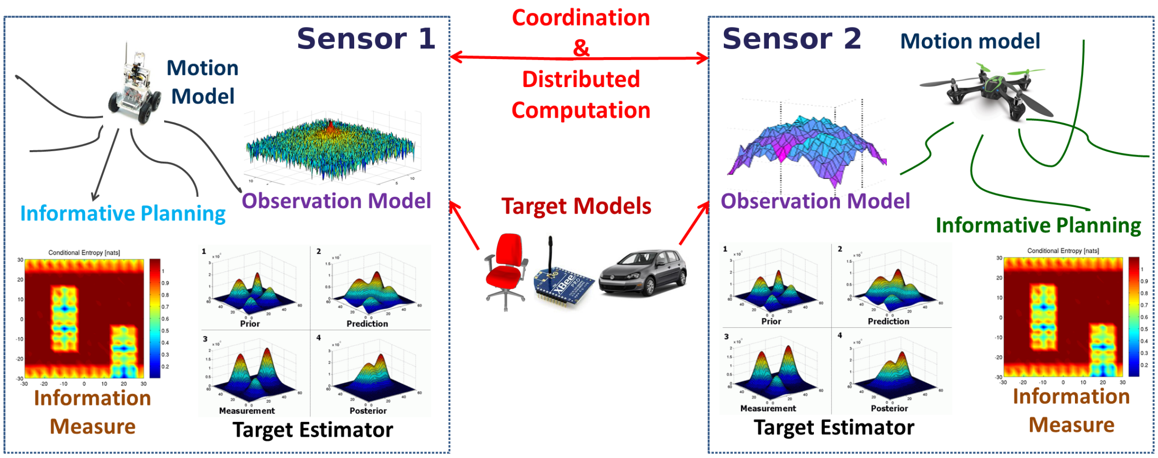


Figure 1.7: To formulate an active information acquisition problem, it is necessary to specify sensor motion models (1.1), target motion models (1.2), sensor observation models (1.3), and an information measure. The task is to design an estimator for the target states and to plan the motion of the sensors in order to improve the estimation performance. The solution should handle heterogeneous sensing systems and should allow for distributed computation and coordination among them.

(Appendix B.5), and an odometry sensor, such as a combination of an inertial measurement unit (IMU) and wheel encoders, can be modeled as a 2-D relative-pose sensor (Appendix B.6) because it estimates the transformation between two successive robot poses. Often times, especially for more complicated systems, the motion and observation models can be learned from training data via machine learning and system identification techniques.

In addition to the sensor and target models described above, there are several other considerations for specifying the active information acquisition problem. We state the problem formally first and discuss the additional ingredients afterwards.

Problem (Active Information Acquisition)**.** *Given initial sensor states x_0, a prior p_0 on the target states, and a planning horizon $T < \infty$, choose control policies $\mu_t := \{\mu_{i,t} \mid i = 1,\dots,n\}$ for $t = 0,\dots,T-1$ that map the history of control inputs and sensor observations to new control inputs for the sensors and aim to maximize information about the target:*

$$\begin{aligned} \max_{\mu_{0:T-1}} \quad & \mathcal{I}(y_{1:T}, x_{1:T}, z_{1:T}) \\ s.t. \;\; x_{t+1} &= f(x_t, \mu_t, noise) \\ y_{t+1} &= a(y_t, noise) \\ z_t &= h(x_t, y_t, noise) \end{aligned} \tag{1.4}$$

where $\mathcal{I}$ is an information measure.

Many variants of the above problem can be considered (e.g., linear or nonlinear models f, a, h; discrete or continuous state and control spaces; various noise distributions and information measures). For example, if $f(x, u, \text{noise}) := u$ and we think of the control u as choosing a subset of sensors to be activated, then (1.4) becomes a sensor scheduling problem (Joshi and Boyd 2009, Vitus et al. 2012). A different type of information gathering problem arises, if the task is to control the motion of a mobile sensor to detect a contamination source or to estimate unknown parameters of spatially-distributed systems (Rafajlowicz 1986). If the maneuverability of the sensor is constrained, the future behavior of the target has to be anticipated at an early stage in order to obtaining good sensor positioning. We discuss the ingredients which determine the type of active information acquisition problem below.

Target state estimator

Before we even consider planning the motion of the sensors, we need to design an estimator which uses sensor observations to refine the knowledge about the target. As stated in the problem formulation, we are given a prior probability density (or mass) function (pdf) p_0 of the target states y_0. The task of the estimator is to propagate the pdf p_t over time by using incoming measurements and the motion and observation models. This is precisely the topic of estimation theory and is addressed by filtering (Huber 2009), smoothing (Kaess 2008), or graphical-model inference (Koller and Friedman 2009) techniques. Appendix C introduces Bayesian filtering and two of the most successful filters in practice - the Kalman filter (Appendix C.1) and the particle filter (Appendix C.3). These filters, along with some more sophisticated variants, are employed for target estimation in this dissertation.

While planning the sensor motion in problem (1.4), a target estimator has to be executed multiple times to predict the target behavior, especially for long planning horizons. Hence, the more accurate and complex the employed estimator is, the more computationally-demanding and informative the resulting sensor control policies are. Here, it is important to note that the estimator employed for informative planning does not necessarily have to be the same as the one used for inferring the target state when the actual sensor measurements arrive. A practical way to trade off estimation accuracy and computational resources is to use a high-fidelty estimator during inference (since it has to be executed only once per time step) and a less-complex estimator during planning (since it has to be executed many times while solving problem (1.4)). This might be suboptimal but in practice inference needs to be precise, while planning needs to be fast!

Information measure

The target state estimator provides a (usually finite-dimensional) representation of the target pdf p_t that can be used to judge the uncertainty (or information) contained in the current estimate of the target state. The information measure $\mathcal{I}(y_{1:T}, x_{1:T}, z_{1:T})$ in problem (1.4) does precisely this - it captures the improvement in the target estimate along different sensor trajectories. For example, we can think of the active information acquisition problem as planning a sensor trajectory that will improve the uncertainty (captured by the covariance matrix) of a Kalman filter the most. Hence, the design of an appropriate information measure is an important consideration for active information gathering. This thesis relies on information theory for suitable candidates such as conditional entropy $\mathbf{H}(y_{1:T}, x_{1:T} \mid z_{1:T})$, mutual information $\mathbf{I}(y_{1:T}, x_{1:T}; z_{1:T})$, and probability of error (see Appendix A for details). The chosen information measure should not only lead to the desired sensor behavior but should also be fast to compute, ideally in closed-form. The reason is that it needs to be evaluated multiple times during the planning process, especially for long planning horizons, and, as mentioned earlier, fast informative planning is requisite in many applications. As we will discover in this dissertation, mutual information has a closed form for a Gaussian target pdf but might be notoriously difficult to evaluate for multimodal distributions (Ch. 3). In the latter case, it might be more appropriate to use a measure such as Cauchy-Schwarz mutual information (Charrow et al. 2015, Kampa et al. 2011) or quadratic Rényi entropy (Rényi 1961, Carrillo et al. 2015a), which have a closed-form expression for mixtures of Gaussians.

Planning horizon

Another determinant of the complexity of the active information acquisition problem is the length T of the planning horizon. If the horizon is $T = 1$, we obtain a (greedy) next-best-sensor-view problem. At the other extreme, if $T = \infty$, we get a persistent monitoring problem, in which attention can be restricted to periodic sensor trajectories (Zhao et al. 2014). In Ch. 2, we will observe that, for $T < \infty$, the complexity of the active information acquisition problem grows exponentially with the length of the planning horizon. Hence, it is desirable to investigate approximate solutions that mitigate this complexity but retain some performance guarantees. Finally, if the planning horizon T is subject to optimization (not fixed), we get an optimal stopping problem, which will be considered in Ch. 4.

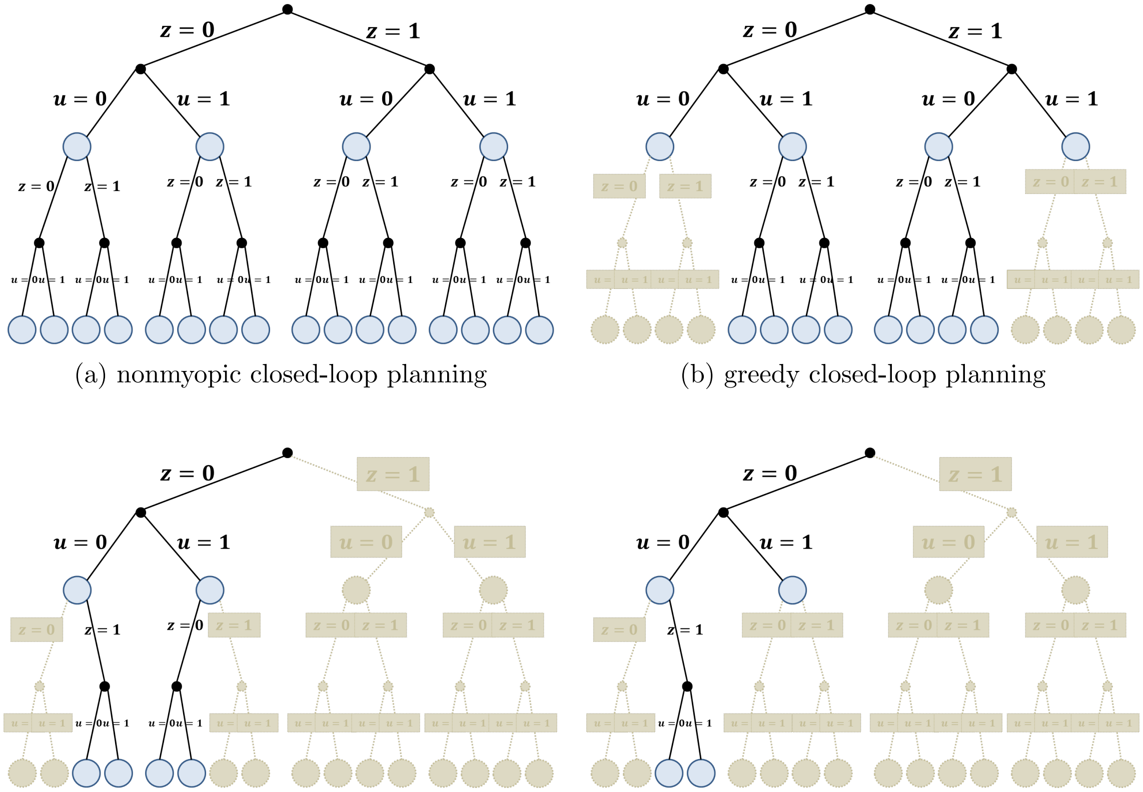
Optimization type

The active information acquisition problem has strong relations to stochastic optimal control for dynamic systems. In fact, in its most general form (1.4) it can be considered an instance of a partially observable Markov decision process (POMDP, Bertsekas (1995)). However, instead of assuming a general value function (as done in POMDP planning), this thesis exploits the properties of the information measure to offer a more scalable, efficient, and precise solution than what is offered by a general POMDP solver.

As mentioned before, one of the factors that determines the complexity of the planning problem is the horizon length T. Perhaps, the easiest instance of the problem is to compute a greedy control policy, i.e., one that quantifies the sensing performance only at the next time step. A solution to a longer-horizon problem can be constructed by committing to the greedy policy in the first step and solving a series of such single-step optimizations. We call this approach *greedy planning.* This is in contrast with *nonmyopic planning* which solves (1.4) directly with a longer planning horizon ($T > 1$).

Planning approaches can also differ in their adaptability to future measurements. *Open-loop planning* fixes some expected values for the future measurements and computes a *sequence* of controls assuming that these most-likely observations will happen. At run time, the same sequence of sensor inputs will be executed regardless of the actual measurement realizations. *Closed-loop* planning, on the other hand, computes a sequence of functions (i.e., a policy) which map to an appropriate control input depending on the actual measurement realizations. As a result, the closed-loop policy is adaptable to the actual measurements at run time.

We can use an example to clarify the distinctions among the four types of optimization: greedy closed-loop, nonmyopic closed-loop, greedy open-loop, and nonmyopic open-loop. Consider a sensing system that can be in two states $x \in \{0, 1\}$ and the control input $u \in \{0, 1\}$ selects the system state without noise. Suppose that the sensor is binary too, $z \in \{0, 1\}$, and measures the target state directly. Let the planning horizon be $T = 2$. To



(a) nonmyopic closed-loop planning

(b) greedy closed-loop planning

(c) nonmyopic open-loop planning

(d) greedy open-loop planning

Figure 1.8: Comparison among greedy, nonmyopic, open-loop, and closed-loop planning variants. The blue nodes represent the sensor states in a tree of depth 2, corresponding to the planning horizon length. Nonmyopic closed-loop planning considers all possible combinations of future observations z and controls u. Nonmyopic open-loop planning, on the other hand, commits to some most-likely observations beforehand and only considers the possible control input variations. The corresponding greedy versions commit to a control policy (or input) at the first level of the tree and then explore only the subtrees starting from it. Intuitively, we can expect nonmyopic closed-loop planning to be the most computationally demanding, while greedy open-loop planning to be the least.

optimize the sensor performance, we need to explore a tree of possible control inputs and measurement realizations of depth $T = 2$. See Fig. 1.8 for a visualization. Nonmyopic closed-loop planning considers all possible control inputs, all possible measurement realizations, and the whole planning horizon (Fig. 1.8(a)). Greedy closed-loop planning, on the other hand, explores a tree of depth 1 first and commits to a single-step control policy. Afterwards, it explores the second level of the tree but, due to committing early, some of the nodes are not reachable (Fig. 1.8(b)). Nonmyopic open-loop planning explores the whole tree but commits to some most-likely observations beforehand. Essentially, it removes all branching possibilities associated with the measurements and then searches the tree for a control sequence, rather than a function of the possible measurements (Fig. 1.8(c)). Finally, greedy open-loop planning commits to some most-likely observations and explores a tree of depth 1 first. It commits to a single control input (not policy) and then explores only the subtree starting from it (Fig. 1.8(d)).

It is clear from Fig. 1.8 that nonmyopic closed-loop planning is the most computationally demanding approach but provides the best sensor performance, since it explores all future possibilities. At the other extreme, greedy open-loop planning is the fastest to compute but might ignore lots of possible future realizations. It has been shown in the literature (Williams 2007, Ch.3.5.1) that greedy closed-loop planning can be arbitrarily worse than nonmyopic closed-loop planning. Greedy closed-loop planning may be competitive with nonmyopic open-loop planning in some settings (Williams 2007, Thm.3.8). Further, nonmyopic open-loop planning may be arbitrarily worse than nonmyopic closed-loop planning (Hollinger et al. 2013a, Thm.1). Interestingly, we will see in Ch. 2 that active information acquisition with linear Gaussian sensing and motion models is an *exception.* In other words, there is no gain in the performance of closed-loop planning over that of open-loop planning. However, we will also show that greedy planning is worse than nonmyopic planning even for static independent targets (Ch. 2).

A compromise between open-loop and closed-loop planning is open-loop feedback control (Bertsekas 1995). Like in open-loop planning, future measurements are fixed at some most-likely values but when an actual measurement arrives it is used for computing an updated plan. This idea of planning an open-loop sequence, applying the first control input, and replanning after the next observation is also known as receding horizon control or model predictive control (Morari and Lee 1999, Rawlings and Mayne 2009). It can be shown that open-loop feedback control is no worse than open-loop planning but again its performance may be arbitrarily worse than that of closed-loop planning (Bertsekas 1995). We will combine nonmyopic open-loop planning with model predictive control in Ch. 2 to obtain an adaptive control policy for nonlinear sensor and target models.

1.3 Related Work

Since many variants of the general formulation in (1.4) have been studied, this section provides only a high-level overview of the related work. The closest approaches will be discussed in more detail within the specific context of the later chapters.

The idea of optimizing the performance of configurable sensing systems dates back to the 60's (Meier et al. 1967, Athans 1972). Earlier approaches to active information acquisition considered stateless sensor systems (systems whose internal state is not affected by the control and all controls are available at all times) in problems such as sensor selection (Joshi and Boyd 2009), sensor placement (Krause 2008), sensor scheduling (Gupta et al. 2006, Le Ny et al. 2011, Vitus et al. 2012), and active hypothesis testing (Naghshvar and Javidi 2013b,a, Nitinawarat and Veeravalli 2013a, Nitinawarat et al. 2013).

More recently, attention has shifted to dynamic sensors and complex estimation techniques but, typically, greedy control. Examples include range-only target tracking (Martínez and Bullo 2006), the next-best-view problem in active object recognition (Bajcsy 1988, Karasev et al. 2012, Denzler and Brown 2002, Sommerlade and Reid 2008, Eidenberger and Scharinger 2010) and "information surfing" (Grocholsky 2002, Schwager et al. 2011), where the sensors follow the gradient of mutual information. Despite the use of greedy control, efficient computation in large sensor teams remains a challenge. Charrow et al. (2013, 2015) proposed deterministic approximations to the classical mutual information and an alternative information measure, which can be computed in closed-form for some measurement distributions. Research has also been focused on distributed information-seeking control

(Hoffmann and Tomlin 2010, Julian et al. 2012, Meyer et al. 2014, Dames and Kumar 2013).

As may be expected, non-greedy active information acquisition is an even harder problem (Le Ny 2008). Ryan and Karl Hedrick (2010) consider the impact of multiple measurements over time in the context of information-theoretic mobile target tracking. Mutual information is approximated using Monte Carlo sampling, which leads to a computationally-expensive algorithm. There is a line of work which alleviates the complexity, associated with longer planning horizons, by assuming linear Gaussian sensing models. Examples include sampling-based (Hollinger and Sukhatme 2013, 2014, Lan and Schwager 2013), search-based (Le Ny and Pappas 2009, Atanasov et al. 2014a) and submodularity-based approaches (Singh et al. 2009a,b) for open-loop informative planning in Gaussian random fields (Le Ny and Pappas 2009) and Gaussian process models (Marchant and Ramos 2014). Finally, there does not exist a specialized information planner for nonmyopic closed-loop information gathering with general sensing models (non-Gaussian and non-linear). However, the problem is an instance of a partially-observable Markov decision process and work in POMDP planning (Kaelbling and Lozano-Pérez 2013, Kurniawati et al. 2008, Silver and Veness 2010) and belief-space planning (Hauser 2011, Indelman et al. 2014, Prentice and Roy 2009) is relevant.

Important applications of active information acquisition include environmental monitoring (Choi 2009, Tokekar 2014), search and rescue (Kumar et al. 2004), security and surveillance (Rybski et al. 2000), structure inspection (Hollinger et al. 2013b), source seeking (Zhang et al. 2007, Ghods and Kristić 2011, Stanković and Stipanović 2010, Atanasov et al. 2015b), active perception (Denzler and Brown 2002, Valente et al. 2014, Atanasov et al. 2014b), and active simultaneous localization and mapping (SLAM, Carlone et al. (2014), Sim and Roy (2005), Kontitsis et al. (2013), Atanasov et al. (2015a)).

1.4 Outline and Contributions

The long-term goals of the work described in this dissertation are:

- **Improved representation**: Before considering controlled information acquisition, it is necessary to establish reliable target modeling and estimation techniques. In most cases, traditional tools such as Gaussian processes, Bayes Nets, and Kalman and particle filters suffice. However, it is desirable to develop inference methods that can simultaneously handle continuous variables (e.g., target poses), discrete variables (e.g., target classes), and sensing nuisances, such as false and missed detections and ambiguous data association (the mapping between received measurements and visible targets).

- **Mature information-planning algorithms**: The main goal is to provide a solution to the active information acquisition problem in (1.4) that is comparable in maturity to traditional planning algorithms such as ARA* (Likhachev et al. 2003) and RRT* (Karaman and Frazzoli 2011). This involves proving theoretic performance guarantees and allowing anytime execution (a valid solution can be provided at any time) and incremental re-planning (information from previous planning episodes is reused). Further, the algorithms should handle heterogeneous sensor and target models (e.g., air and ground vehicles, linear and nonlinear sensing models, discrete and continuous measurements, static and mobile targets).

- **Scalability**: Both the inference and the information-planning approaches need to scale well with respect to the length of the planning horizon and the number of sensing systems. This inevitably leads to considerations for distributed estimation and control techniques that facilitate coordination and collaboration among the sensing systems.
- **Experimental validation**: The proposed algorithms should be validated in real-world experiments and should be compared with alternative approaches.

Below is a brief overview of the chapters and their contributions with respect to the existing work and the goals listed above.

Chapter 2: Dynamic optimization over a long time horizon is required to obtain a sensor control policy that anticipates the target evolution and avoids local minima of the information measure. To deal with the increasing complexity of longer planning horizons, this chapter takes advantage of two key observations:

- An active information acquisition problem with nonlinear non-Gaussian motion and observation models can be converted into a linear Gaussian one via statistical linearization as long as the motion and observation models are continuous.
- The classical separation principle (Athans 1972, Meier et al. 1967) between estimation and control holds for the linear Gaussian Active Information Acquisition problem. As a result, the optimal estimator is the Kalman filter and the problem can be reduced from a stochastic optimal control problem to a deterministic one, where open-loop policies achieve the optimal solution. Still, the complexity of computing the optimal nonmyopic open-loop policy scales exponentially with the planning horizon T and the number of sensors n.

The chapter makes the following contributions.

- We prove the separation principle holds for the linear Gaussian Active Information Acquisition problem.
- We develop an approximation algorithm with finite-time suboptimality guarantees for nonmyopic open-loop planning that manages the complexity in T. The algorithm can be used in combination with model predictive control to produce an adaptive policy for nonlinear motion and observation models.
- We propose a decentralized control scheme to achieve linear complexity in the number of sensors n. We prove that the approach obtains at least 50% of the (mutual) information obtained by the optimal centralized solution.
- In addition to the control task, we decentralize the target inference process. We develop a distributed Kalman filter for dynamic target tracking and a distributed Jacobi algorithm for sensor self-localization. We prove that the two algorithms can be used in conjunction to achieve joint localization and estimation in sensor networks with arbitrarily small asymptotic mean square error.

- We present applications in methane emission monitoring, mobile vehicle tracking, distributed localization in sensor networks, and active multi-robot simultaneous localization and mapping.

Chapter 3: In the presence of sensing nuisances, such as missed detections and false alarms, and discrete or strongly nonlinear observation models, the conversion into a linear Gaussian information gathering problem is of limited fidelity. Even worse, sometimes the target motion model and sensor observation model are completely unknown. The classical estimation techniques, such as the Kalman filter and the particle filter, are not sufficient here. The focus of this chapter is on target inference techniques that can deal with such difficult situations. Due to the complexity of the inference process, the control in this chapter is greedy. We make the following contributions.

- We develop a sensor observation model for set-valued measurements which incorporates missed detections, false alarms, and data association.

- We prove that obtaining the likelihood of a set-valued sensor observation is equivalent to a matrix permanent computation. This allows a polynomial-time approximation to a Bayes filter with set-valued observations.

- We develop a stochastic gradient ascent scheme for multi-sensor information gathering with unknown sensor observation and target motion models. Gaussian radial basis functions are used to calculate finite-difference weights which approximate the gradient of the measured signal. We prove that our approach converges to a local maximum of the measured signal.

- We present applications in vehicle localization in residential areas using semantically meaningful landmarks, global localization of Google's Project Tango phone (Google ATAP group 2014), active robot localization using object detections, and wireless radio source localization.

Chapter 4: The simplest case in which the linearization technique of Ch. 2 cannot be applied is when some measurement or state variables are discrete. For example, we face such complications in active vision problems where scene labels and object classes are discrete. Unfortunately, removing the linear Gaussian assumptions invalidates the separation principle and closed-loop planning is needed to solve the original stochastic optimal control problem (1.4). Unlike Ch. 3 which uses greedy planning, this chapter focuses on nonmyopic closed-loop planning for information gathering in discrete state and measurement spaces. The chapter makes the following contributions.

- We formulate an active information acquisition problem with motion and observation models described by arbitrary probability mass functions. This formulation is particularly suitable for active object recognition, classification, and hypothesis testing problems.

- We develop an exact nonmyopic closed-loop planning algorithm via dynamic programming (Bertsekas 1995). The algorithm obtains the optimal solution but scales poorly with the size of the state and measurement spaces. To provide scalability, we also propose an approximate algorithm based on Monte Carlo tree search (Browne et al. 2012) with a rollout policy that exploits the structure of the information measure.

- We demonstrate that these ideas can improve object classification and pose estimation with a mobile camera and can also speed up object recognition with deformable part models (Felzenszwalb et al. 2010b).

The thesis closes with conclusions and possible extensions in Ch. 5.

Chapter 2

Nonmyopic Information Acquisition with Linear Gaussian Models

Sensor management (Hero III and Cochran 2011) offers a formal methodology to control the degrees of freedom of sensing systems in order to improve the information acquisition process. The earliest approaches date back to the 60's (Meier et al. 1967, Athans 1972). Efficient nonmyopic planning approaches have been proposed for sensors without internal states in the context of sensor placement and scheduling (Gupta et al. 2006, Le Ny et al. 2011, Krause et al. 2008, Krause 2008, Vitus et al. 2012, Williams 2007, Joshi and Boyd 2009). However, when it comes to optimizing the trajectories of mobile sensors, which possess internal states, the approaches are often greedy and rarely provide performance guarantees. The main complication is that the evolution of the sensor states depends on the control inputs and affects the measurements in the long run. Concretely, whereas in radar management a sensor can switch instantaneously between targets (Krishnamurthy and Evans 2001), a feasible and informative path needs to be designed for a sensing robot (Singh et al. 2009a). Due to this complication, most approaches for information gathering with mobile sensors are greedy (Grocholsky 2002, Chung et al. 2006) or use short planning horizons (Kreucher 2005, Huber 2009). However, it is precisely the presence of internal states that makes multi-step optimization important. The behavior of the observed phenomenon needs to be predicted at an early stage to facilitate effective control of the mobile sensor.

This chapter exploits a key insight that under linear Gaussian assumptions on the target and observation models the active information acquisition problem (1.4) can be formulated as a deterministic optimal control problem (Le Ny and Pappas 2009). As a result, informative sensing paths can be computed via open-loop planning and there is no loss in performance with respect to closed-loop planning. The advantage is that the search space of a deterministic control problem is significantly smaller than the stochastic version. The problem can be solved forward in time, considering reachable states only, instead of discretizing the high-dimensional search space as required by backward value iteration (Bertsekas 1995). Still, the complexity of the optimal nonmyopic open-loop solution turns out to be exponential in both the planning horizon T and the number of sensors n.

To manage the complexity in the planning horizon, this chapter develops an approximation algorithm with strong performance guarantees. The key idea is to discard future sensor trajectories that are close in space and dominated in informativeness by other trajectories from the search space. We prove that this results in bounded suboptimality with respect to the optimal centralized solution and allows much faster nonmyopic planning. Coupled with linearization and model predictive control, this approach can even generate adaptive policies for non-linear motion and observation models. Our work can be considered a search-based method for planning in information space. Related work in this area includes (Zhou 2012, Vander Hook et al. 2014, Le Ny and Pappas 2009). Some successful approaches for open-loop informative planning rely on a *submodular* function to quantify the informativeness of the sensor paths (Singh et al. 2009a,b). The sensing locations are partitioned into independent clusters. Submodularity is used to greedily select informative locations within clusters. An orienteering problem is solved to choose the sequence of clusters to visit. The drawback is that within clusters the movement of the sensor is restricted to a graph, essentially ignoring the sensor dynamics. As a result, the cluster sizes cannot be increased much without affecting the quality of the solution. We address this limitation by considering the sensor dynamics and planning non-greedily. Sampling-based methods have been proposed as well (Lan and Schwager 2013, Hollinger and Sukhatme 2013). Just as in traditional planning, they are able to find feasible solutions quickly but provide no finite-time guarantees on optimality. Approaches which do not make linear Gaussian assumptions and use nonlinear filters exist as well (Charrow et al. 2013, Dames et al. 2012, Hoffmann and Tomlin 2010, LaValle 2012, Julian et al. 2012, Meyer et al. 2014, Lauri and Ritala 2014). Charrow et al. (2013) choose trajectories based on a fast and accurate approximation of the mutual information between nonlinear Gaussian measurements and a particle representation of the target distribution. Dames et al. (2012) and Meyer et al. (2014) have the sensors follow the gradient of mutual information and conditional entropy, respectively. Lauri and Ritala (2014) use Monte Carlo tree search to solve a finite-horizon stochastic control problem with MI as the reward. A common theme in these works is that the length of the planning horizon is sacrificed in favor of using the (typically) nonlinear sensor and target models during planning. We take the opposite stance and give up some model accuracy (via linearization) in order to plan efficiently with long horizons. An additional benefit is that any sparsity in the target information matrix can be exploited during the planning process. Recent advances in large-scale finite and infinite dimensional estimation (e.g. Gaussian Processes (Quiñonero-Candela and Rasmussen 2005, Barfoot et al. 2014), graph-based SLAM (Dellaert and Kaess 2006, Kaess et al. 2012)) take advantage of such structure inherent in a graphical-model representation of the environment. Since the information planning process is based on the estimation layer, it is natural that it utilizes a sparse representation too.

To address the scalability in the number of sensors, instead of solving the information acquisition problem jointly over all sensors, we propose a decentralized control scheme based on coordinate descent, which proceeds as follows. The first sensor plans its own trajectory, without taking the others into account, and passes the information over to the second sensor. The second sensor plans its own trajectory by solving a two-sensor active information acquisition problem but with a known control sequence for the first sensor. Then, sensor 2 passes its information along to sensor 3, which solves a 3-sensor problem with known control sequences for the first two sensors and so on. In addition to achieving linear complexity in the number of sensors, our approach obtains at least 50% of the (mutual) information obtained by an optimal centralized solution. To prove this property we rely on the fact that mutual information, when viewed as a function of the sensor set, is submodular. To alleviate the computational burden of multi-robot deployments, other approaches in the literature also decentralize the information-seeking control task. Hoffmann and Tomlin (2010) compute mutual information only for pairs of sensors, thus decreasing the dimension of the required integration. Similarly, Dames et al. (2012) achieve scalability by assuming that mutual information approximately decouples among robot cliques. Fully-distributed approaches based on belief consensus are proposed by Julian et al. (2012) and Meyer et al. (2014). Decentralization based on coordinate descent has been used by Singh et al. (2009a) and a similar approach based on Gauss-Seidel Relaxation was proposed by Zhou and Roumeliotis (2011). Of these works, only Singh et al. (2009a) provide guarantees on the performance of their approach, valid for static targets and discrete sensing locations, without revisiting. Here, we allow revisiting and extend the guarantees to dynamic targets.

Finally, in many applications it is necessary to decentralize the target inference process in addition to the control task. To complicate matters, it is often the case that the sensors need to know their own locations with respect to a common reference in order to utilize the target measurements meaningfully. Hence, they face a *joint* localization and estimation problem. Existing work in distributed target estimation assumes implicitly that the localization problem is solved, while all the literature on localization does not consider the effect of the residual errors on a common estimation task. Sec. 2.6 develops a distributed Kalman filter for dynamic target tracking and a distributed Jacobi algorithm for sensor self-localization. We prove that the two algorithms can be used in conjunction with arbitrarily small asymptotic mean square error. Our distributed target tracking algorithm was inspired by Rahnama Rad and Tahbaz-Salehi (2010), who propose an approach for distributed *static* parameter estimation using nonlinear sensing models. We specialize their model to heterogeneous sensors with linear Gaussian observations, derive the update step of the resulting linear estimator, and show stronger convergence results (mean-square consistency instead of weak consistency). Finally, we add a prediction step to the filter in order to handle dynamic targets. Our filter is similar to the Kalman-Consensus filter (Olfati-Saber 2007, 2009) and the filter proposed by Khan et al. (2010) and Shahrampour et al. (2013) but uses different gains in the update step and the performance guarantees (for static targets) we obtain are stronger. Khan et al. (2010) show that their filter can track a dynamic target with bounded error if the norm of the target system matrix is less than the network tracking capacity. Shahrampour et al. (2013) quantify the filter's estimation performance using a global loss function and show that the asymptotic error depends on the loss function decomposition. Kar et al. (2012) study distributed static parameter estimation with nonlinear observation models and noisy inter-sensor communication. Related work also includes Cortés (2009), which combines the Jacobi over-relaxation method with dynamic consensus to compute distributed weighted least squares.

Our localization algorithm follows the lines of the Jacobi algorithm, first proposed for localization in sensor networks by Barooah and Hespanha (2007), Barooah (2007). In contrast to their approach, we consider repeated relative measurements and show strong convergence guarantees for the resulting sequential algorithm. Other work in sensor network localization typically considers nonlinear and less informative measurement models. For instance Aspnes et al. (2006), Moore et al. (2004), So and Ye (2007), Priyantha et al. (2003)

address range-only localization, which is challenging because a graph with specified edge lengths can have several embeddings in the plane. Khan et al. (2009) introduce a distributed localization algorithm, which uses the barycentric coordinates of a node with respect to its neighbors and show convergence via a Markov chain. Diao et al. (2013) relax the assumption that all nodes must be inside the convex hull of the anchors. Localization has also been considered in the context of camera networks (Tron and Vidal 2009). None of the papers mentioned above, however, consider state estimation and localization jointly. While the joint problem can be cast as a large distributed estimation task and addressed with existing methods (e.g. the Kalman-Consensus filter (Olfati-Saber 2007, 2009)), the resulting solution will have each sensor estimate the locations of *all* other sensors in the network. This has slow convergence and is infeasible for large networks as it requires repeated exchange of information that scales with the size of the network. The proposed joint algorithm resolves this challenge and in addition keeps the update rules linear.

This chapter is based on the papers (Atanasov et al. 2014a, 2015a, 2014c). It presents applications in methane emission monitoring, mobile vehicle tracking, and multi-robot active SLAM.

2.1 Linear Gaussian Active Information Acquisition

The active information acquisition problem (1.4) is challenging when we consider nonmyopic planning. One approach to alleviate the complexity associated with longer planning horizons is to linearize the sensor and target models and use model predictive control. In this chapter, we will show that a version of the celebrated separation principle (Athans 1972, Meier et al. 1967) between estimation and control holds for the active information acquisition problem under the following assumptions.

- The target motion model is linear in the target state and the noise is Gaussian.
- The sensor observation model is linear in the target state and the noise is Gaussian. The model can still depend nonlinearly on the sensor state.
- The sensor motion model is deterministic.
- The information measure has the form: $\mathcal{I}(y_{1:T}, x_{1:T}, z_{1:T}) := \sum_{t=1}^{T} c_t(\Sigma_t, x_t)$, where $\Sigma_t := \mathbb{E}(y_t - \mathbb{E}y_t)(y_t - \mathbb{E}y_t)^T$ is the target covariance matrix and c_t is a per-stage reward. For example, the mutual information between Gaussian random variables can be written in this form (Williams 2007).

As mentioned earlier, these assumptions will allows us to restrict attention to open-loop planning, which is significantly less demanding than closed-loop planning. We state the linear Gaussian version of the active information acquisition problem formally. Let $\mathcal{X}_i \cong \mathbb{R}^{d_{xi}}$ be the state space of sensor i and let $\mathcal{U}_i$ be a *finite* space of admissible control inputs (e.g., motion primitives). Also, let $\mathcal{U} := \mathcal{U}_1 \times \ldots \times \mathcal{U}_n$ and $\mathcal{X} := \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ with associated metric $d_{\mathcal{X}}$. The information available to the sensors at time t to compute the control inputs $u_t \in \mathcal{U}_t$ is:

$$\mathcal{H}_0 := (x_0, z_0), \quad \mathcal{H}_t := (x_{0:t}, z_{0:t}, u_{0:(t-1)}),\ t > 0.$$

Then, a *control policy* for sensor i is a sequence of functions $\mu_{i,t}(\mathcal{H}_t) \in \mathcal{U}_i$ for $t = 0, \ldots, T-1$ that maps a history $\mathcal{H}_t$ of control inputs, sensor states, and sensor observations to a control input in the set of admissible controls $\mathcal{U}_i$.

Problem (Linear Gaussian Active Information Acquisition)**.** *Given initial sensor states $x_0 \in \mathcal{X}$, a prior on the target states y_0, and a finite planning horizon T, choose control policies $\mu_t := \{\mu_{i,t} \in \mathcal{U}_i \mid i = 1, \ldots, n\}$ for $t = 0, \ldots, T-1$ that minimize the differential entropy (see Appendix A) of the target states conditioned on the measurement set $z_{1:T}$:*

$$\begin{aligned} \min_{\mu_{0:T-1}} \quad & \mathbf{h}(y_T \mid z_{1:T}) \quad OR \quad \sum_{t=1}^{T} \mathbf{h}(y_t \mid z_{1:t}) \\ s.t. \;\; & x_{t+1} = f(x_t, \mu_t) \\ & y_{t+1} = A_t(x_t, u_t) y_t + w_t, \quad w_t \sim \mathcal{N}(0, W_t(x_t, u_t)) \\ & z_t = H_t(x_t) y_t + v_t, \quad v_t \sim \mathcal{N}(0, V_t(x_t)) \end{aligned} \tag{2.1}$$

where v_t is a sensor-state-dependent Gaussian measurement noise, whose values are independent at any pair of times and across sensors, and w_t is uncorrelated Gaussian noise on the target process, which is independent of the measurement noise v_t but might depend on the sensor states x_t and the control inputs u_t.

In addition to the linear Gaussian assumptions on the target motion model and the sensor observation model, we notice two other differences with respect to the general problem (1.4) in Sec. 1.2. First, we have chosen conditional differential entropy as a particular measure of uncertainty (here we minimize uncertainty instead of maximizing information). This choice will be discussed in Sec. 2.3. We consider two types of cost functions: a terminal-stage-only cost function and an additive stage cost. The first one is concerned only with the performance at the terminal time, while the second one minimizes the uncertainty in the target state along the whole sensor trajectory. Depending on the application, either cost function might be effective. For example, in localization problems with high-speed vehicles it is not desirable to have growing uncertainty at any time so the second cost function would be more suitable. However, for search-and-rescue problems we might want to increase the uncertainty while getting to a critical region and the first cost function would be more appropriate.

The second difference from problem (1.4) is that the sensor motion model is deterministic (i.e., not corrupted by noise), while the target motion model is subject to control. The idea behind this choice is to separate the deterministic evolution of some sensor properties from the uncertain dynamics. This simplifies the planning process but does not limit the applicability of the models. In particular, even if the sensor transitions are nondeterministic, the evolution of the (deterministic) mean sensor state can be captured in x_t, while the (nondeterministic) deviation from the mean can be captured in the (now controlled) target state y_t. For example, in Sec. 2.5.1 we will see an application to active SLAM, in which the (deterministic) mean of the robot pose estimates will be captured by x_t, while the (nondeterministic) error in those estimates will be captured by y_t.

Finally, note that problem (2.1) is still a *multi-sensor multi-target* problem with sensor states $x_t := [x_{1,t}^T \; \cdots \; x_{n,t}^T]^T \in \mathcal{X}$. To emphasize its dependence on the number of sensors n

and the planning horizon length T, we use the following shorthand notation:

$$\min_{\mu_{1:n,0:(T-1)}} J_T^{(n)}\left(\mu_{1,0:(T-1)}, \ldots, \mu_{n,0:(T-1)}\right). \tag{2.2}$$

2.2 Separation Principle and Optimal Centralized Solution

When the models are linear Gaussian and the target prior is Gaussian, the optimal target estimator is the Kalman filter (Appendix C.1). The target distribution remains Gaussian over time and its covariance evolves according to the Riccati map and very importantly is independent of the particular realization of the measurement sequence. As a result, the differential entropy (Appendix A) of the Gaussian target distribution conditioned on the Gaussian measurements is proportional to the log-determinant of the target covariance matrix. In other words, *the cost function in* (2.1) *is independent of the particular measurement realization and, consequently, open-loop planning does just as well as closed-loop planning.* The following theorem formalizes this intuition.

Theorem 2.1. *If the distribution of y_0 is Gaussian with covariance $\Sigma_0 \succeq 0$, there exists an open-loop control sequence $\sigma \in \mathcal{U}^T$ that is optimal in (2.1). Moreover, (2.1) is equivalent to the deterministic optimal control problem:*

$$\begin{aligned} \min_{\sigma \in \mathcal{U}^T} \; & J_T^{(n)}(\sigma) := \sum_{t=1}^{T} c_t(\Sigma_t, x_t) \\ s.t. \;\; & x_{t+1} = f(x_t, \sigma_t), && t = 0, \ldots, T-1, \\ & \Sigma_{t+1} = \rho_{t+1}^e(\rho_t^p(\Sigma_t, x_t, \sigma_t), x_{t+1}), && t = 0, \ldots, T-1, \end{aligned} \tag{2.3}$$

where $c_T(\Sigma_T, x_T) := \log\det(\Sigma_T)$ and for $t = 1, \ldots, T-1$:

$$c_t(\Sigma_t, x_t) := \begin{cases} 0 & \textit{for the terminal-stage-only cost in } (2.1) \\ \log\det(\Sigma_t) & \textit{for the additive stage cost in } (2.1) \end{cases}$$

Further, $\rho_t^p(\Sigma, x, u)$ is the Kalman filter covariance prediction:

$$\rho_t^p(\Sigma, x, u) := A_t(x, u)\Sigma A_t^T(x, u) + W_t(x, u) \tag{2.4}$$

and $\rho_t^e(\Sigma, x)$ is the Kalman filter covariance update:

$$\begin{aligned} \rho_t^e(\Sigma, x) &:= \Sigma - \Sigma H_t^T(x)\left(H_t(x)\Sigma H_t^T(x) + V_t(x)\right)^{-1} H_t(x)\Sigma \\ &= (\Sigma^{-1} + M_t(x))^{-1} \end{aligned} \tag{2.5}$$

*where $M_t(x) := H_t^T(x)V_t^{-1}(x)H_t(x)$ is called the **sensor matrix**.*

Notation: In the reminder we suppress the dependence on x and u of A_t, W_t, H_t, V_t, $\rho_t^p(\Sigma)$, $\rho_t^e(\Sigma)$, and M_t when it is clear from context in order to simplify the notation. We also define $R_t(\Sigma) := I - K_t(\Sigma)H_t$ and the Kalman gain $K_t(\Sigma) := \Sigma H_t^T(H_t\Sigma H_t^T + V_t)^{-1}$. Further, given a control sequence $\sigma := u_\tau, \ldots, u_{t-1} \in \mathcal{U}^{t-\tau}$ at time τ, the corresponding sensor trajectory is denoted $\pi(\sigma) := x_\tau, \ldots, x_t \in \mathcal{X}^{T-\tau+1}$.

One computational bottleneck in problem (2.3) is the large dimension of the state (x_t, Σ_t). The significance of the separation principle[1] (Thm. 2.1) is that search space can be explored forward in time by building a set of reachable states rather than discretizing the high-dimensional space of all possible covariance as required by a closed-loop approach such as backward value iteration (Bertsekas 1995). As shown by Le Ny and Pappas (2009), the optimal (nonmyopic) open-loop control sequence σ^* in (2.3) can be obtained via forward value iteration (FVI, Alg. 1). FVI constructs a *search tree*, $\mathcal{T}_t$, with nodes at stage $t \in [0, T]$ corresponding to the reachable states (x_t, Σ_t, J_t). At each node, there are edges, one for each control in $\mathcal{U}$, leading to nodes $(x_{t+1}, \Sigma_{t+1}, J_{t+1})$, obtained from the dynamics in (2.3). Unfortunately, FVI has exponential complexity, $O(|\mathcal{U}_1 \times \ldots \times \mathcal{U}_n|^T)$, in both the time horizon T and the number of sensors n, since the number of nodes in $\mathcal{T}_t$ equals the number of sensor trajectories of length t.

Algorithm 1 Forward Value Iteration

```
$J_0 \leftarrow 0, \quad S_0 \leftarrow \{(x_0, \Sigma_0, J_0)\}, \quad S_t \leftarrow \emptyset$ for $t = 1, \ldots, T$
for $t = 1 : T$ do
    for all $(x, \Sigma, J) \in S_{t-1}$ do
        for all $u \in \mathcal{U}$ do
            $x_t \leftarrow f(x, u), \quad \Sigma_t \leftarrow \rho^e_{t+1}(\rho^p_t(\Sigma, x, u), x_t)$
            $J_t \leftarrow J + c_t(\Sigma_t, x_t)$
            $S_t \leftarrow S_t \cup \{(x_t, \Sigma_t, J_t)\}$
return $\min \{J \mid (x, \Sigma, J) \in S_T\}$
```

The other extreme is greedy open-loop planning, which discards all but the best (lowest cost) node at level t of the tree $\mathcal{T}_t$. The greedy policy σ^g is

$$\sigma^g_t \in \underset{u \in \mathcal{U}}{\arg\min} \left(\log\det \left(\rho^e_{t+1}(\rho^p_t(\Sigma_t, x_t, u), f(x_t, u)) \right) \right), \; t \in [0, T-1] \tag{2.6}$$

and has linear complexity in the length of the planning horizon, $\mathcal{O}(|\mathcal{U}_1 \times \ldots \times \mathcal{U}_n|T)$, but no guarantees exist for its performance. Fig. 2.1 shows a graphical comparison between greedy and nonmyopic planning in the context of problem (2.3). A natural question is if a solution that does less work than FVI but has suboptimality guarantees can be found. Our idea is to approximate FVI by discarding some nodes from the search tree. Unlike the greedy approach which discards everything except the currently-best node, we should discard nodes more intelligently in order to retain some performance guarantees. Before we proceed, however, we need to be sure that the greedy policy is not optimal.

Example 2.1. Consider the following single-sensor linear Gaussian active information acquisition problem with planning horizon $T = 2$.

$$\begin{aligned} \min_{\sigma \in \{1,2,3\}^2} \;& \mathbf{h}(y_2 \mid z_{1:2}) & \\ \text{s.t. } \;& x_{t+1} = \sigma_t, & t = 0, 1, \\ & y_{t+1} = y_t, \quad y_0 \sim \mathcal{N}(0, I_2) & t = 0, 1, \\ & z_t = H(x_t) y_t + v_t, \quad v_t \sim \mathcal{N}(0, V(x_t)) & t = 1, 2, \end{aligned}$$

[1] Note that Thm. 2.1 guarantees the ability to plan open-loop which in general is stronger than the separation principle. For example, linear quadratic Gaussian regulation satisfies the separation principle but open-loop regulation cannot be achieved. Here, the cost function (2.3) is crucial in guaranteeing independence from the particular measurement realizations.

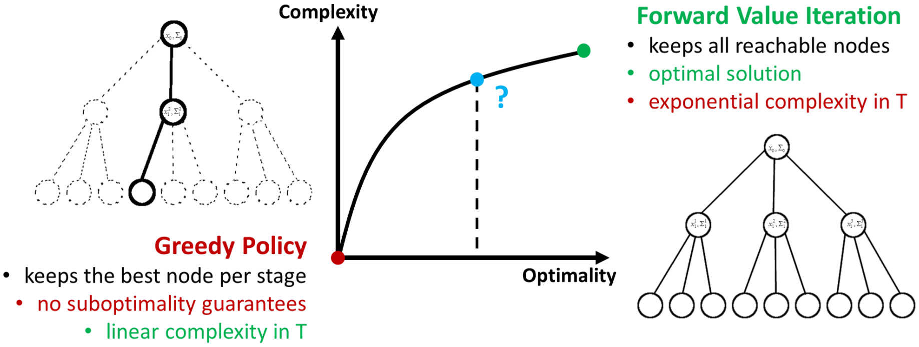

Complexity
?
Optimality
Forward Value Iteration
• keeps all reachable nodes
• optimal solution
• exponential complexity in T
Greedy Policy
• keeps the best node per stage
• no suboptimality guarantees
• linear complexity in T

Figure 2.1: Forward value iteration (FVI) is a nonmyopic open-loop planning approach which constructs a search tree (right) with branching factor $|\mathcal{U}|$ and depth T. It is guaranteed to find the optimal control sequence σ^* in (2.3) but its complexity is exponential in T (and n). Greedy open-loop planning, on the other hand, keeps only the best node per stage (left) and, hence, has linear complexity in T (and exponential in n) but provides no performance guarantees.

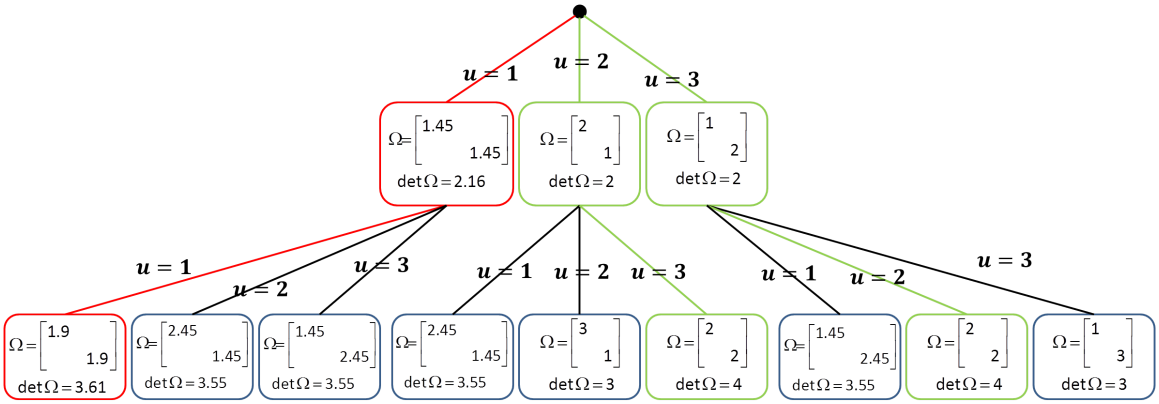


Figure 2.2: Example that greedy planning is worse than nonmyopic planning even for static independent targets and a planning horizon of $T = 2$. The control sequence chosen by greedy planning is indicated in red, while the optimal two-step sequences are shown in green.

where the sensor observation model is defined by the sensor matrix $M(x) := H^T(x)V^{-1}(x)H(x)$ as follows:

$$M(1) = \begin{bmatrix} 0.45 & 0 \\ 0 & 0.45 \end{bmatrix} \quad M(2) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad M(3) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Let $\Omega_t := \Sigma_t^{-1}$ be the target information matrix at time t. Due to the separation principle (Thm. 2.1), the problem is equivalent to:

$$\begin{aligned} \max_{\sigma \in \{1,2,3\}^2} \quad & \log \det(\Omega_2) \\ \text{s.t.} \quad & \Omega_{t+1} = \Omega_t + M(\sigma_t), \quad t = 0, 1. \end{aligned}$$

Fig. 2.2 shows the search tree of depth 2 needed to compute the optimal open-loop policy and the control inputs chosen by the greedy policy. We can see that the greedy policy commits to a suboptimal input at the first stage and hence achieves a worse reward than the optimal control sequence. It is noteworthy that greedy planning is not optimal even in such a simple setting with static independent targets. Interestingly, this example also shows that the optimal control policy is not unique.

2.3 On the Choice of Cost Function

Before we proceed with an approximation of FVI, we discuss the choice of conditional differential entropy as a cost function in problem (2.1). The theory of optimal designs (Kiefer 1974) proposes several efficiency measures (functionals of the eigenvalues of the target information matrix $\Omega_t := \Sigma_t^{-1}$), which can be used to quantifying the performance of the target estimator. The most widely-used information measures are A-optimality ($-\operatorname{tr}(\Omega_t^{-1})$), D-optimality ($\det(\Omega_t)$), T-optimality ($\operatorname{tr}(\Omega_t)$), and E-optimality ($\lambda_{\min}(\Omega_t)$). Information theory (Cover and Thomas 2012) provides another set of functions, such as mutual information and conditional entropy (see Appendix A for details), which can be used to quantify the uncertainty in the target state given the received measurements.

Without offering an exhaustive treatment of the subject, we present some evidence that conditional entropy (proportional to D-optimality in the case of Gaussian variables) is superior to the other information measures for the purposes of active information acquisition. The first indication is provided by Carrillo et al. (2015b). The authors carry out a robot pose estimation experiment in which the robot receives no measurements and uses only its motion model to estimate the pose over time (dead reckoning). It is natural to expect that the uncertainty of the robot pose estimate should grow monotonically when there is no measurement feedback. However, the authors show that certain circular robot motions may cause the A-opt and E-opt criteria to decrease. D-opt is the only criterion that does not exhibit this undesirable behavior. A second indication that conditional entropy is a superior criterion can be observed when the target transitions are controlled (i.e., the target transition matrix A is a function of u as in (2.1)). Since the sensor motion model is assumed deterministic in (2.1), controlled target transitions are needed to capture any uncertainty in the sensor dynamics (see the active SLAM application in Sec. 2.5.1 for more details). Table 2.1 below shows that mutual information might be inadequate as a value function in this case. The conditional differential entropy criterion is more appropriate (note that the two are equivalent if the target transitions are not controlled).

Table 2.1: Suppose that the sensor state is x_0, the target covariance is $\Sigma_0 = 1$, and there are two available controls $u^{(1)}, u^{(2)}$. The table shows an example in which the target state is a lot more uncertain after $u^{(1)}$ than after $u^{(2)}$ but nevertheless $u^{(1)}$ is considered more informative by the mutual information value function. The problem arises because, while entropy measures the uncertainty in absolute terms, mutual information measures the change in uncertainty after the prediction step. As a result, mutual information encourages controls that create a lot of uncertainty if it can then be reduced significantly via the measurement. In the SLAM context, mutual information might prefer very uncertain (e.g., high velocity) controls even if they provide the same measurement information (captured by the sensor matrix $M(x)$ below) as more certain ones.

	$u^{(1)}$	$u^{(2)}$
$A(x_0, u)$	1	1
$W(x_0, u)$	99999	1
$M(x_1) := H^T(x_1)V^{-1}(x_1)H(x_1)$	0.01	0.01
$\Sigma_{1\|0} := \rho_0^p(\Sigma_0, x_0, u) = A\Sigma_0 A^T + W$	100000	2
$\Sigma_1 = \rho_1^e(\Sigma_{1\|0}, x_1) = ((\Sigma_{1\|0})^{-1} + M(x_1))^{-1}$	99.9	1.96
$\mathbf{h}(y_1 \mid z_1) := \frac{1}{2}(\log(2\pi e) + \log\det(\Sigma_1))$	3.72	**1.76**
$\mathbb{I}(y_1; z_1) := \mathbf{h}(y_1) - \mathbf{h}(y_1 \mid z_1)$	**3.45**	0.01

2.4 Managing Complexity due to the Planning Horizon

The goal of this section is to provide an algorithm, with complexity lower than that of forward value iteration (Alg. 1) and performance better than that of the greedy policy (2.6). This can be achieved by discarding some but not all of the nodes at level t of the search tree $\mathcal{T}_t$. In short, we define notions of sensor trajectories being δ-close in space and ϵ-close in informativeness and prove that if several sensor trajectories δ-cross at time t and one is ϵ-dominated in informativeness by the rest, then it can be discarded from the future search with minimal loss in performance.

2.4.1 Optimality-preserving Reductions

Intuitively, if two sensor trajectories cross, i.e., there are two nodes at level t of $\mathcal{T}_t$ with the same sensor state x but different target covariances, and one is clearly less informative, i.e., its uncertainty about the target is larger (in the positive semidefinite sense) than that of the other one, then it is not necessary to keep it. In the same vein, the covariance of a sensor path at time t may be dominated by a combination of the covariances of several other paths. The following definition makes this notion precise.

Definition 2.1 (Algebraic redundancy (Vitus et al. 2012))**.** Let $\{\Sigma_i\}_{i=1}^K$ be a finite set of symmetric positive semidefinite matrices. A symmetric matrix $\Sigma \succeq 0$ is algebraically redundant with respect to $\{\Sigma_i\}$, if there exist nonnegative constants $\{\alpha_i\}_{i=1}^K$ such that $\sum_{i=1}^K \alpha_i = 1 \quad \text{and} \quad \Sigma \succeq \sum_{i=1}^K \alpha_i \Sigma_i$.

The next theorem shows that when several sensor paths cross at time t, the algebraically redundant ones can be discarded without removing the optimal one under the following assumption.

Assumption. *The stage cost $c_t(\Sigma, x)$ is concave and monotone in Σ.*

Theorem 2.2 (Optimal reduction)**.** *For $t \in [1, T]$, let $(x, \Sigma, J) \in S_t$ be a node in level t of the search tree $\mathcal{T}_t$. If there exist a set $\{x^i, \Sigma^i, J^i\} \subseteq S_t \setminus \{(x, \Sigma, J)\}$ such that $x = x^i$, $\forall i$ and $\mathbf{diag}(\Sigma, J)$ is algebraically redundant with respect to $\{\mathbf{diag}(\Sigma^i, J^i)\}$, then (x, Σ, J) can be removed from S_t without eliminating the optimal trajectory.*

Remark. If a terminal-stage-only cost is used, i.e., $c_t(\Sigma, x) = 0$ for $t < T$, then it is not necessary to keep track of the running cost J_t. It is sufficient to check Σ with respect to $\{\Sigma^i\}$ for algebraic redundancy.

2.4.2 ϵ-Suboptimal Reductions

At the expense of losing optimality, we can discard even more of the crossing trajectories by using a relaxed notion of algebraic redundancy.

Definition 2.2 (ϵ-Algebraic redundancy (Vitus et al. 2012))**.** Let $\epsilon \geq 0$ and let $\{\Sigma_i\}_{i=1}^K$ be a finite set of symmetric positive semidefinite matrices. A symmetric matrix $\Sigma \succeq 0$ is ϵ-algebraically redundant with respect to $\{\Sigma_i\}$, if there exist nonnegative constants $\{\alpha_i\}_{i=1}^K$ such that $\sum_{i=1}^K \alpha_i = 1 \quad \text{and} \quad \Sigma + \epsilon I \succeq \sum_{i=1}^K \alpha_i \Sigma_i$.

Denote the search tree at time t with all ϵ-algebraically redundant nodes removed by $\mathcal{T}_t^\epsilon$. Let $\pi^\epsilon = x_0, x_1^\epsilon, \ldots, x_T^\epsilon \in \mathcal{X}^{T+1}$ be the trajectory obtained by forward value iteration on the reduced tree $\mathcal{T}_T^\epsilon$ with a corresponding covariance sequence $\Sigma_0, \Sigma_1^\epsilon, \ldots, \Sigma_T^\epsilon$ and cost J_T^ϵ. To quantify its performance we focus on the terminal-stage-only cost in (2.1), i.e, $J_T^\epsilon := \log\det(\Sigma_T^\epsilon)$. Let $\pi^* = x_0, x_1^*, \ldots, x_T^* \in \mathcal{X}^{T+1}$ be the optimal sensor trajectory in $\mathcal{T}_T$ with covariance sequence $\Sigma_0, \Sigma_1^*, \ldots, \Sigma_T^*$ and cost $J_T^* := \log\det(\Sigma_T^*)$. The following theorem provides an upper bound on the gap, $(J_T^\epsilon - J_T^*)$, between the performances of π^ϵ and π^*.

Theorem 2.3 (ϵ-Suboptimal reduction)**.** *For $t \in [1, T]$, let $\beta_* < \infty$ be the peak estimation error of the optimal policy, i.e., $\Sigma_t^* \preceq \beta_* I$ and $A_t^* \Sigma_t^* A_t^* + W_t^* \preceq \beta_* I$, and let $\lambda_* > 0$ be a lower bound on the information obtained per step along the optimal trajectory, i.e., $\lambda_* I \preceq \left((W_t^*)^{-1} + M_{t+1}^*\right)^{-1}$. Then,*

$$0 \le J_T^\epsilon - J_T^* \le \epsilon\Delta, \qquad \Delta := \frac{d_y}{\lambda_*}\left(1 + \frac{\beta_*^5}{\lambda_*^5}\right) \tag{2.7}$$

where d_y is the target dimension.

Remark. If the peak estimation error β_* remains bounded as $T \to \infty$, i.e., the sensors performs well when using the optimal policy, then $\Delta < \infty$. In other words, the suboptimality gap of π^ϵ is bounded regardless of the length T of the planning horizon and grows linearly in ϵ!

2.4.3 (ϵ, δ)-Suboptimal Reductions

If the sensor motion is restricted to a graph, many of the possible future trajectories will be intersecting and the results from Thm. 2.3 are very satisfactory. However, if the space of sensor configurations, $\mathcal{X}$, is continuous, depending on the sensor motion model (e.g., differential drive), it is possible that no two sensor states reachable at time t are exactly equal. Then, the reductions from Thm. 2.3 cannot be applied. To avoid this case, we can relax the notion of trajectory crossing at time t.

Definition 2.3 (Trajectory δ-Crossing)**.** Trajectories $\pi^1, \pi^2 \in \mathcal{X}^{T+1}$ δ-cross at time $t \in [1, T]$ if $d_{\mathcal{X}}(\pi_t^1, \pi_t^2) \le \delta$ for $\delta \ge 0$.

Instead of discarding ϵ-algebraically redundant trajectories which cross, we can discard those which δ-cross. Let $\mathcal{T}_t^{\epsilon,\delta}$ be the reduced tree at time t, with all ϵ-algebraically redundant δ-crossing nodes removed. Some continuity assumptions are necessary in order to provide suboptimality guarantees for searching in $\mathcal{T}_t^{\epsilon,\delta}$.

Assumption 2.1 (**Motion Model Continuity**)**.** *The sensor motion model is Lipschitz continuous in x with Lipschitz constant $L_f \ge 0$ for every fixed $u \in \mathcal{U}$:*

$$d_{\mathcal{X}}(f(x_1, u), f(x_2, u)) \le L_f d_{\mathcal{X}}(x_1, x_2), \quad \forall x_1, x_2 \in \mathcal{X}.$$

Assumption 2.2 (**Target and Observation Model Continuity**)**.** *Consider arbitrary $t \ge 0$, $u \in \mathcal{U}$, $x_1, x_2 \in \mathcal{X}$, and $\Sigma \succeq 0$. Let $\rho_i(\Sigma) := \rho_{t+1}^e(\rho_t^p(\Sigma, x_i, u), f(x_i, u))$ for $i = 1, 2$. Assume that the matrices $A_t(x, u)$, $W_t(x, u)$, $H_t(x)$, and $V_t(x)$ in (2.1) are defined so that* *there exists a real constant $L_m \geq 0$ such that for $0 < \gamma := (1 + L_m d_{\mathcal{X}}(x_1, x_2))^{-1} \leq 1$, the Riccati map satisfies the following continuity notion:*

$$\rho_1(\Sigma) \succeq \gamma\rho_2(\Sigma) + (1-\gamma)\rho_1(0),$$
$$\rho_2(\Sigma) \succeq \gamma\rho_1(\Sigma) + (1-\gamma)\rho_2(0).$$

Assumption 2.1 says that when two sensor configurations are close and the same sequence of controls is applied, then the resulting trajectories will remain close. Assumption 2.2 says that sensing from similar configurations results in similar information gain. This gives the intuition that when two trajectories δ-cross their future informativeness will be similar. We make this intuition precise in the next theorem. Let $\pi^{\epsilon,\delta} \in \mathcal{X}^{T+1}$ be the sensor trajectory obtained by searching the reduced tree $\mathcal{T}_T^{\epsilon,\delta}$ with corresponding cost $J_T^{\epsilon,\delta} := \log\det(\Sigma_T^{\epsilon,\delta})$. The gap, $(J_T^{\epsilon,\delta} - J_T^*)$, between the performances of $\pi^{\epsilon,\delta}$ and π^* is bounded as follows.

Theorem 2.4 ((ϵ, δ)-Suboptimal reduction)**.** *For $t \in [1, T]$, let $\beta_* < \infty$ be the peak estimation error of the optimal policy, i.e., $\Sigma_t^* \preceq \beta_* I$ and $A_t^*\Sigma_t^* A_t^* + W_t^* \preceq \beta_* I$, and let $\lambda_* > 0$ be a lower bound on the information obtained per step along the optimal trajectory, i.e., $\lambda_* I \preceq \left((W_t^*)^{-1} + M_{t+1}^*\right)^{-1}$. Then,*

$$0 \leq J_T^{\epsilon,\delta} - J_T^* \leq (\zeta_T - 1)\left[J_T^* - \log\det(\lambda_* I)\right] + \epsilon\Delta_T, \tag{2.8}$$

where $\zeta_t := \prod_{\tau=1}^{t-1}\left(1 + \sum_{s=1}^{\tau} L_f^s L_m \delta\right) \geq 1$, $\Delta_T := \frac{d_y}{\lambda_*}\left(1 + \frac{\beta_*}{\lambda_*}\sum_{\tau=1}^{T-1}\frac{\zeta_T}{\zeta_\tau}\eta_*^{T-\tau}\right)$,
$\eta_* := \frac{\beta_*^4}{\beta_*^4 + \lambda_*^4} < 1$, and d_y is the target dimension.

Notice that Thm. 2.3 is a special case of Thm. 2.4 because if $\delta = 0$, then $\zeta_t = 1, \forall t \in [1, T]$ and (2.8) reduces to (2.7). Then, the suboptimality gap remains bounded regardless of the time horizon and grows linearly with ϵ. If $\delta > 0$, then $\lim_{T\to\infty}\zeta_T = \infty$ and the suboptimality gap grows with T and δ. The bound is loose, however, because it uses a worst-case analysis. The worst case happens when a trajectory, which was discarded from $\mathcal{T}_T^{\epsilon,\delta}$, persistently obtains more information than a trajectory, which remains very close in space and is still in the search tree. Even then, if the optimal policy performs well, the term $J_T^* - \log\det(\lambda_* I)$ gets smaller as ζ_T increases and the suboptimality gap remains small.

2.4.4 (ϵ, δ)-Reduced Value Iteration

The approaches for reducing the search tree, developed in the previous subsections, can be used to significantly decrease the complexity of the FVI algorithm (Alg. 1), while providing suboptimality guarantees (Thm. 2.4). The (ϵ, δ)-Reduced Value Iteration (RVI) is summarized in Alg. 2. Note again that if a terminal-stage-only cost is used, i.e., $c_t(\Sigma, x) = 0$ for $t < T$, then it is not necessary to keep track of the running cost J_t and is sufficient to check if Σ is algebraically redundant.

An appealing property of Alg. 2 is that at stage t at least the node with currently lowest cost is retained (Line 8). This guarantees that the control sequence obtained from RVI *performs at least as well as* the greedy policy (2.6). RVI is a powerful technique because

Algorithm 2 (ϵ, δ)-Reduced Value Iteration

```
 1: $J_0 \leftarrow 0, \quad S_0 \leftarrow \{(x_0, \Sigma_0, J_0)\}, \quad S_t \leftarrow \emptyset$ for $t = 1, \ldots, T$
 2: for $t = 1 : T$ do
 3:     for all $(x, \Sigma, J) \in S_{t-1}$ do
 4:         for all $u \in \mathcal{U}$ do
 5:             $x_t \leftarrow f(x, u), \quad \Sigma_t \leftarrow \rho^e_{t+1}(\rho^p_t(\Sigma, x, u), x_t)$
 6:             $J_t \leftarrow J + c_t(\Sigma_t, x_t)$
 7:             $S_t \leftarrow S_t \cup \{(x_t, \Sigma_t, J_t)\}$
 8:     Sort $S_t$ in ascending order according to $\max\{J_t, c_T(\Sigma_t, x_t)\}$
 9:     $S'_t \leftarrow S_t[1]$                                        ▷ State with lowest cost
10:     for all $(x, \Sigma, J) \in S_t \setminus S_t[1]$ do
11:         % Find all nodes in $S'_t$, which $\delta$-cross $x$:
12:         $Q \leftarrow \{\mathbf{diag}(\Sigma', J') \mid (x', \Sigma', J') \in S'_t,\ d_{\mathcal{X}}(x, x') \leq \delta\}$
13:         if isempty($Q$) or not( $\mathbf{diag}(\Sigma, J)$ is $\epsilon$-alg. red. wrt $Q$ ) then
14:             $S'_t \leftarrow S'_t \cup (x, \Sigma, J)$
15:     $S_t \leftarrow S'_t$
16: return min $\{J \mid (x, \Sigma, J) \in S_T\}$
```

it provides a trade-off between complexity and optimality which can be controlled via the parameters (ϵ, δ). In particular, if both parameters are zero, we obtain the optimal solution with exponential complexity in T, while if both are infinite, we obtain the greedy solution with linear complexity in T, which keeps only the best node at level t of the tree $\mathcal{T}_t$.

The bottleneck of Alg. 2 is checking ϵ-algebraic redundancy (Line 13), which requires solving a linear matrix inequality (LMI) feasibility problem in d_y dimensions. It is a convex feasibility problem and off-the-shelf solvers exist (MATLAB 2012). However, if a solution for a short planning horizon is needed quickly (e.g., in applications with nonlinear models, it is necessary to re-linearize the models and re-plan the trajectories after every few measurements), the algebraic redundancy can be verified only approximately, i.e., for a finite number of values for $\{\alpha_i\}_{i=1}^K$. In particular, choosing $\alpha_j = 1$ and $\alpha_i = 0, i \neq j$ for $j = 1, \ldots, K$, requires verifying only K positive-semidefinite inequalities:

$$\Sigma + \epsilon I \succeq \Sigma_i, \quad i = 1, \ldots, K, \tag{2.9}$$

instead of solving the LMI feasibility problem. As a result, some dominated nodes might not be removed but this does not affect the performance guarantees in Thm. 2.2, Thm. 2.3, and Thm. 2.4.

2.4.5 Exploiting Sparsity

A fact that does not play a role in the problem formulation but is important for the solution is that if the interactions among the target states are local (e.g. static landmarks), the *information matrix*, $\Omega_0 := \Sigma_0^{-1}$, is *sparse*, whereas the covariance matrix is not (Le Ny and Pappas 2009). For example, the information structure in SiLAM is naturally sparse (Dellaert and Kaess 2006), since it encodes constraints from measurements that connect only two variables (two robot poses or a robot pose and a landmark location). Since the planning process (2.3) uses prior information from the estimation layer, it is natural that it utilizes a sparse representation too. Yet, Alg. 2 does not exploit this.

The sparsity of Ω_t can be leveraged by replacing the Kalman filter in Alg. 2 with a square-root information filter (SRIF, Bierman (1974)). We take a closer look at the target motion model and decompose the state y_t into a static part (y^s) and dynamic part (y^d):

$$\begin{aligned} y^d_{t+1} &= A^d_t(x_t, u_t) y^d_t + w^d_t, \quad w^d_t \sim \mathcal{N}(0, W^d_t(x_t, u_t)),\\ y^s_{t+1} &= y^s_t, \end{aligned} \tag{2.10}$$

so that $A_t(x, u) := \mathbf{diag}(A^d_t(x, u), I)$ and $W_t(x, u) := \mathbf{diag}(W^d_t(x, u), 0)$. Since most existing SRIF formulations require invertible noise covariances, we derive SRIF equations that specifically handle the case that part of the target state is static and the associated covariance is zero. Let the Cholesky factor (informally, square root[2]) of the information matrix be the upper triangular matrix C_t, such that $\Omega_t = C_t^T C_t$, and decompose it into blocks:

$$C_t = \begin{bmatrix} D_t & F_t \\ 0 & G_t \end{bmatrix}, \tag{2.11}$$

where D_t corresponds to the dynamic target states (y^d_t) and G_t - to the static target states (y^s_t).

Proposition 2.5. *The square-root information filter equations, with prior square-root matrix C_t in* (2.11)*, process model in* (2.10)*, measurement model in* (2.1)*, and known x_t, x_{t+1}, u_t, are:*

$$\textbf{\textit{Predict:}} \quad \begin{bmatrix} (W^d_t)^{-1/2} & 0 & 0 \\ D_t(A^d_t)^{-1} & D_t(A^d_t)^{-1} & F_t \\ 0 & 0 & G_t \end{bmatrix} = Q \begin{bmatrix} * & * \\ 0 & C_{t+1|t} \end{bmatrix}$$

$$\textbf{\textit{Update:}} \quad \begin{bmatrix} C_{t+1|t} \\ V^{-1/2}_{t+1} H_{t+1} \end{bmatrix} = \bar{Q} \begin{bmatrix} C_{t+1} \\ 0 \end{bmatrix}$$

$$\textbf{\textit{Combined:}} \quad \begin{bmatrix} (W^d_t)^{-1/2} & 0 & 0 \\ D_t(A^d_t)^{-1} & D_t(A^d_t)^{-1} & F_t \\ 0 & 0 & G_t \\ 0 & V^{-1/2}_{t+1} H_{t+1} \end{bmatrix} = \hat{Q} \begin{bmatrix} * & * \\ 0 & C_{t+1} \\ 0 & 0 \end{bmatrix}$$

where $Q, \bar{Q}, \hat{Q}$ are orthogonal matrices such that the other matrices on the right-hand sides are upper triangular. The dependence of A^d_t, W^d_t on x_t, u_t and of H_{t+1}, V_{t+1} on x_{t+1} is not explicit to simplify notation.

Proposition 2.5 states that the update and prediction steps can be computed via QR decomposition. To use the SRIF and avoid computing the covariance matrices in the reduced value iteration algorithm (Alg. 2), we express the cost c_t and the approximate algebraic redundancy condition (2.9) in terms of C. The cost becomes:

$$c_t(\Sigma, x) = \log\det(\Sigma) = -\log\big(\det(C)^2\big) = -2\textstyle\sum_{i=1}^{d_y} \log|C_{ii}|,$$

[2] The Cholesky factorization of a positive definite matrix can be interpreted as its square root but it is not necessarily the same as the unique nonnegative square root of the matrix.

while (2.9) can be modified via a Schur complement:

$$
\begin{aligned}
\Omega_i - (\Sigma + \epsilon I)^{-1} &\succeq 0, && i = 1, \dots, K,\\
C_i^T C_i - C^T (I + \epsilon C C^T)^{-1} C &\succeq 0, && i = 1, \dots, K,\\
\begin{bmatrix} I + \epsilon C C^T & C \\ C^T & C_i^T C_i \end{bmatrix} &\succeq 0, && i = 1, \dots, K.
\end{aligned}
$$

2.4.6 Linearization and Model Predictive Control

In this section, we will show that the solution we developed for the linear Gaussian active information acquisition problem, coupled with linearization and model predictive control, can generate an adaptive policy for the general active information acquisition problem (1.4) with nonlinear models. The only requirement is that the nonlinear models can be linearized (differentiability is sufficient but a high-fidelity linear approximation can be obtained even without it in some cases (Huber 2009, 2015)). In particular, consider the general active information acquisition problem:

$$
\begin{aligned}
\max_{\mu_{0:T-1}} \quad & \mathbf{h}(x_T, y_T \mid z_{1:T}) \\
\text{s.t.} \quad & x_{t+1} = \mathsf{f}(x_t, \mu_t, \eta_t), \qquad \mathbb{E}\eta_t = 0,\ \mathbb{E}\eta_t\eta_t^T = E_0 \\
& y_{t+1} = a(y_t, w_t), \qquad \mathbb{E}w_t = 0,\ \mathbb{E}w_t w_t^T = W_0 \\
& z_t = h(x_t, y_t, v_t), \qquad \mathbb{E}v_t = 0,\ \mathbb{E}v_t v_t^T = V_0
\end{aligned}
\tag{2.12}
$$

where η_t, w_t, v_t are noise terms with finite second moments and we assume a prior on the sensor and target states with mean $(\hat{x}_0^T,\ \hat{y}_0^T)^T$ and covariance Σ_0 is given. For $t \in [0, T-1]$ and some fixed control sequence $u_0, \dots, u_{T-1} \in \mathcal{U}^T$, define the predicted trajectory of the (deterministic) mean $(\hat{x}_0^T,\ \hat{y}_0^T)^T$ of the joint sensor-target state and the predicted measurements as follows:

$$
\begin{aligned}
\hat{x}_{t+1} &= \mathsf{f}(\hat{x}_t, u_t, 0)\\
\hat{y}_{t+1} &= a(\hat{y}_t, 0)\\
\hat{z}_{t+1} &= h(\hat{x}_{t+1}, \hat{y}_{t+1}, 0).
\end{aligned}
$$

Further, let $\delta x_t := x_t - \hat{x}_t$, $\delta y_t := y_t - \hat{y}_t$, and $\delta z_t := z_t - \hat{z}_t$ be the (nondeterministic) prediction errors. Assuming that f, a, h are differentiable, we can linearize them around the predicted trajectories:

$$
\begin{aligned}
\delta x_{t+1} &\approx \left[\frac{\partial \mathsf{f}}{\partial x}(\hat{x}_t, u_t, 0)\right] \delta x_t + \left[\frac{\partial \mathsf{f}}{\partial \eta}(\hat{x}_t, u_t, 0)\right] \eta_t\\
\delta y_{t+1} &\approx \left[\frac{\partial a}{\partial y}(\hat{y}_t, 0)\right] \delta y_t + \left[\frac{\partial a}{\partial w}(\hat{y}_t, 0)\right] w_t\\
\delta z_{t+1} &\approx \left[\frac{\partial h}{\partial x}(\hat{x}_{t+1}, \hat{y}_{t+1}, 0) \quad \frac{\partial h}{\partial y}(\hat{x}_{t+1}, \hat{y}_{t+1}, 0)\right] \begin{pmatrix} \delta x_{t+1} \\ \delta y_{t+1} \end{pmatrix} + \left[\frac{\partial h}{\partial v}(\hat{x}_{t+1}, \hat{y}_{t+1}, 0)\right] v_t
\end{aligned}
$$

If these approximate models are used in place of the original nonlinear models in (2.12), we obtain a linear Gaussian active information acquisition problem and can solve it via the techniques developed in this chapter. In particular, we get an instance of problem (2.3) with sensor state $(\hat{x}_t^T, \hat{y}_t^T)^T$, target state $(\delta x_t^T, \delta y_t^T)^T$ and models defined as follows:

$$
\begin{aligned}
f(\hat{x}, \hat{y}, u) &:= \begin{bmatrix} \mathsf{f}(\hat{x}, u, 0) \\ a(\hat{y}, 0) \end{bmatrix} \\
A(\hat{x}, \hat{y}, u) &:= \begin{bmatrix} \frac{\partial \mathsf{f}}{\partial x}(\hat{x}, u, 0) & 0 \\ 0 & \frac{\partial a}{\partial y}(\hat{y}, 0) \end{bmatrix} \\
W(\hat{x}, \hat{y}, u) &:= \begin{bmatrix} \left[\frac{\partial \mathsf{f}}{\partial \eta}(\hat{x}, u, 0)\right] E_0 \left[\frac{\partial \mathsf{f}}{\partial \eta}(\hat{x}, u, 0)\right]^T & 0 \\ 0 & \left[\frac{\partial a}{\partial w}(\hat{y}, 0)\right] W_0 \left[\frac{\partial a}{\partial w}(\hat{y}, 0)\right]^T \end{bmatrix} \\
H(\hat{x}, \hat{y}) &:= \begin{bmatrix} \frac{\partial h}{\partial x}(\hat{x}, \hat{y}, 0) & \frac{\partial h}{\partial y}(\hat{x}, \hat{y}, 0) \end{bmatrix} \\
V(\hat{x}, \hat{y}) &:= \left[\frac{\partial h}{\partial v}(\hat{x}, \hat{y}, 0)\right] V_0 \left[\frac{\partial h}{\partial v}(\hat{x}, \hat{y}, 0)\right]^T
\end{aligned}
$$

As mentioned earlier, higher-fidelity linearizations are possible and since (2.1) allows for time-varying matrices in the models, we can use very precise linear versions of the equations in (2.12). Another important note is that the presence of noise in the original sensor dynamics in (2.12) does not prevent us from using the techniques of this chapter.

The linearization quality depends on the predicted target trajectory, which in turn depends on the measurements obtained online. As a result, it is necessary to re-plan the sensor trajectories online after every new measurement. In general, re-planning would be feasible only for a short horizon $T < T_{max}$ before the plan is needed. Alg. 3 shows how to use the (ϵ, δ)-RVI and model predictive control to generate an adaptive policy. We emphasize

Algorithm 3 Model predictive control via the (ϵ, δ)-RVI

```
1: Input: $T_{max}, \hat{x}_0, \hat{y}_0, \Sigma_0, \mathcal{U}, \mathsf{f}, a, h, E_0, W_0, V_0, T, \epsilon, \delta$
2: for $t = 0 : T_{max}$ do
3:     Predict a target trajectory of length $T$: $\hat{y}_t, \ldots, \hat{y}_{t+T}$
4:     Linearize the models $(\mathsf{f}, a, h, E_0, W_0, V_0)$ to get $f$, $\{A_\tau, W_\tau, H_\tau, V_\tau\}_{\tau=0}^T$
5:     Plan a sensor trajectory of length $T$:
6:         $\sigma \leftarrow (\epsilon, \delta)\text{-RVI}(\hat{x}_t, \hat{y}_t, \Sigma_t, f, \mathcal{U}, \{A_\tau, W_\tau, H_\tau, V_\tau\}_{\tau=0}^T, T)$
7:     Move the sensors: $x_{t+1} \leftarrow \mathsf{f}(x_t, \sigma_1, \eta_t)$
8:     Get a new observation: $z_{t+1} \leftarrow h(x_{t+1}, y_{t+1}, v_{t+1})$
9:     Update the sensor and target estimates: $(\hat{x}_{t+1}, \hat{y}_{t+1}, \hat{\Sigma}_{t+1}) \leftarrow Filter(\hat{x}_t, \hat{y}_t, \hat{\Sigma}_t, z_{t+1})$
```

that the linearized motion and observation models are utilized merely for determining the next set of control inputs (Line 6), while the state inference (Line 9) can still be performed with *any nonlinear filter*.

2.4.7 Application: Methane Emission Monitoring

To motivate the discussion so far, we consider a methane emission monitoring problem, originally addressed by the Best Service Robotics paper (Hernandez Bennetts et al. 2013) at ICRA 2013. The task is to estimate the gas concentration in a landfill using a remote methane leak detector (RMLD) based on tunable diode laser absorption spectroscopy (Fig.

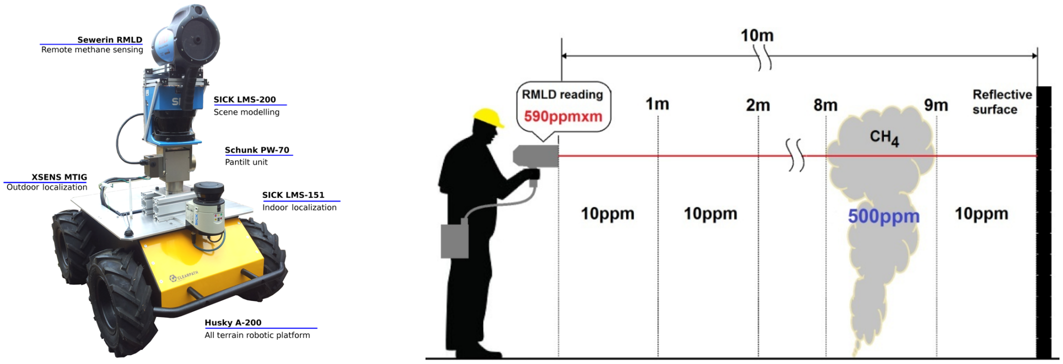


Figure 2.3: Mobile robot equipped with a remote methane leak detector (RMLD) sensor based on tunable diode laser absorption spectroscopy. The RMLD sensor returns a gas concentration measurement in parts-per-million (ppm) integrated along the distance in meters (m) traveled by the laser beam (courtesy of Hernandez Bennetts et al. (2013)).

2.3). The RMLD sensor is mounted on a robotic platform, Gasbot, which follows an exploration path *pre-specified by hand*. The techniques proposed in this chapter can automatically generate the most informative trajectory for the Gasbot.

In detail, suppose that state of the Gasbot consists of its 2D position $(x^1, x^2) \in \mathbb{R}^2$ and the orientation of the RMLD sensor $\theta \in SO(2)$ so that $x := (x^1, x^2, \theta)^T$. At each time period, the Gasbot can move on a grid and choose the orientation of its sensor in 30° increments: $\Theta := \{-\pi, -5\pi/6, \ldots, 5\pi/6\}$,

$$\mathcal{U} := \left\{ (u^1, u^2, u^3)^T \,\middle|\, \begin{pmatrix} u^1 \\ u^2 \end{pmatrix} \in \{0, \pm e_1, \pm e_2\}, u^3 \in \Theta \right\},$$

where e_1 and e_2 are the standard basis vectors in $\mathbb{R}^2$. With $u_t \in \mathcal{U}$, the sensor motion model is:

$$x_{t+1} = f(x_t, u_t) := (x_t^1 + u_t^1,\ x_t^2 + u_t^2,\ u_t^3)^T.$$

Given a time horizon T, we would like to choose a control policy for the Gasbot in order to obtain a good map $y_T \in \mathbb{R}^{d_y}$ of the gas concentration in the landfill. The i-th component, y_T^i, represents the estimate of the concentration in parts per million (ppm) in the i-th cell of the map. We assume a static methane field so that $A = I_{d_y}, W = 0$. It was verified experimentally by Hernandez Bennetts et al. (2013) that a good sensor observation model is:

$$z_t = H(x_t)y_t + v_t = \sum_{i=1}^{d_y} l_i(x_t) y_t^i + v_t, \quad v_t \sim \mathcal{N}(0, V),$$

where the i-th component of $H(x_t) \in \mathbb{R}^{1 \times d_y}$ is the distance $l_i(x_t)$ traveled by sensor laser beam in cell y_t^i for the given sensor pose x_t. Solving problem (2.1) will provide an automatic and informative way to control the Gasbot in order to obtain an accurate map of the methane concentration. We apply the (ϵ, δ)-RVI algorithm (Alg. 2).

Since the movement of the Gasbot is restricted to a grid, the possible sensor trajectories will be crossing frequently and we can use $\delta = 0$. The dimension d_y of the target is the number of cells in the gas concentration map and would typically be very large. Checking

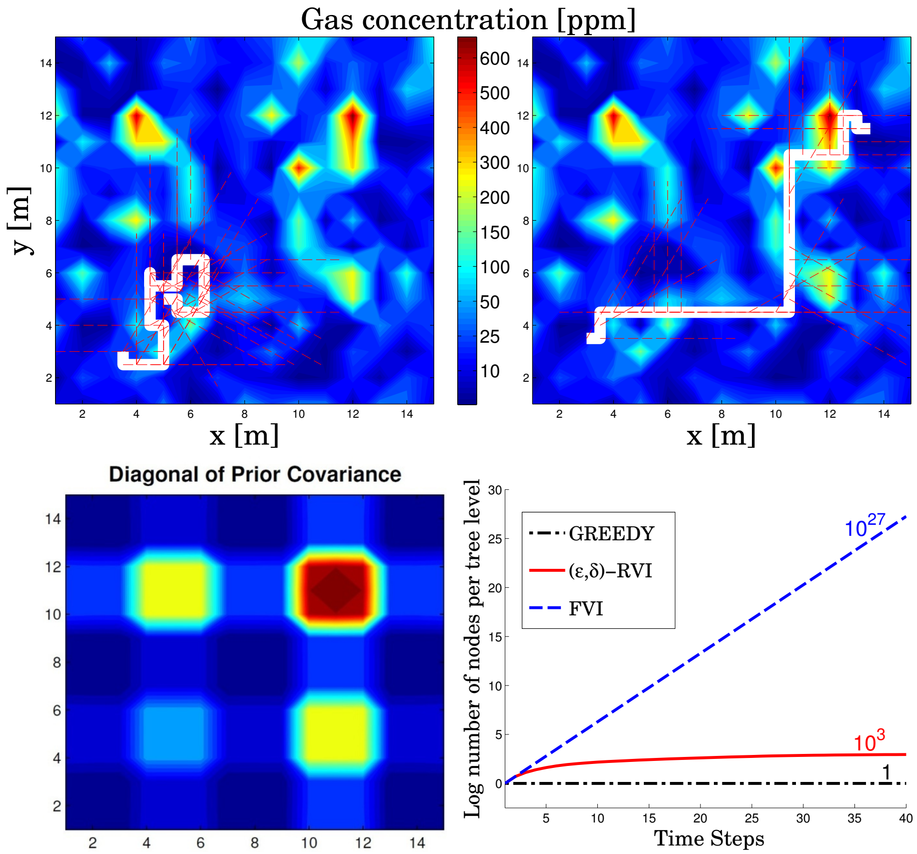


Figure 2.4: Comparison of the sensor trajectories (white) obtained by the greedy algorithm (top left) and the reduced value iteration algorithm (top right) with $\epsilon = \infty$ and $\delta = 0$ after 40 time steps. A typical realization of the methane field is shown on top, while the diagonal of the prior covariance matrix is shown on the bottom left. The red lines indicate the orientation of the gas sensor during the execution. On the bottom right, the log number of nodes maintained in the search tree by the two approaches is compared to the complete tree maintained by forward value iteration.

algebraic redundancy requires solving an d_y-dimensional LMI feasibility problem, which is computationally very demanding. To avoid this, we let $\epsilon = \infty$ (alternatively, we could use the approximate redundancy verification in (2.9)). This means that when several paths cross at time t, only the most informative one is kept in $\mathcal{T}_t^{\epsilon,\delta}$. Thus, the number of nodes in $\mathcal{T}_t^{\epsilon,\delta}$ remains bounded by the number of reachable sensor states. Trajectories of length $T = 40$ were planned using RVI and the greedy policy (2.6, GREEDY). The results (Fig. 2.4) reveal an important drawback of GREEDY. It remains trapped in a local region of relatively high variance and fails to see that there are more interesting regions which should be explored during the limited available time. This observation resembles the outcome we already saw in example 2.1. Fig. 2.4 also shows that the number of nodes maintained in the search tree grows much slower with RVI than with forward value iteration, while the quality of the RVI solution is better than the greedy one. We can conclude that the parameters (ϵ, δ) allow us to manage the complexity in T and choose the operating point on the complexity-optimality curve shown in Fig. 2.1. In particular, if both parameters are zero, we obtain the optimal solution with exponential complexity in T, while if both are

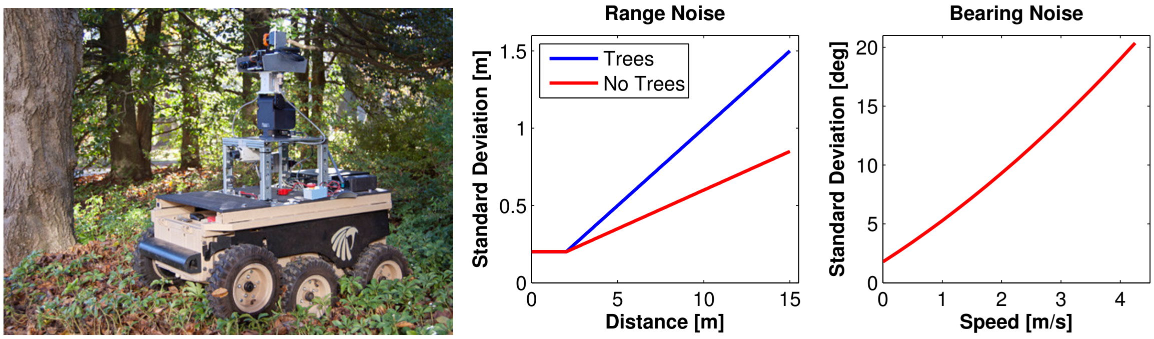


Figure 2.5: The Landshark differential-drive robot (left) tracks a mobile target through a wooded area via range and bearing measurements. The plots on the right show the dependence of the sensor performance on the robot speed, the distance to the target, and the target visibility, which can be obstructed by trees.

infinite, we obtain the greedy solution with linear complexity in T.

2.4.8 Application: Mobile Vehicle Tracking

In this section, we apply the (ϵ, δ)-RVI algorithm in a mobile vehicle tracking application that involves nonlinear models. Suppose that the sensor is mounted on a vehicle with position $(x_t^1, x_t^2) \in \mathbb{R}^2$, orientation $\theta_t \in SO(2)$, and differential-drive dynamics discretized with a sampling period $\tau = 0.5$ s as described in Appendix B.1. The vehicle is controlled using the motion primitives $\mathcal{U} = \{(\nu, \omega) \mid \nu \in \{0, 1, 2, 3\}$ m/s, $\omega \in \{0, \pm\pi/2, \pm\pi\}$ rad/s$\}$. The task is to track the position $(y_t^1, y_t^2) \in \mathbb{R}^2$ and velocity $(\bar{y}_t^1, \bar{y}_t^2) \in \mathbb{R}^2$ of another vehicle with double integrator dynamics, discretized with a sampling period $\tau = 0.5$ s and driven by Gaussian noise with diffusion strength $q = 0.2$ m^2/s^3. The target motion model is described in Appendix B.2. The sensor takes noisy range and bearing measurements of the target's position:

$$h(x_t, y_t) := \begin{bmatrix} \sqrt{(y_t^1 - x_t^1)^2 + (y_t^2 - x_t^2)^2} \\ \arctan\big((y_t^2 - x_t^2)/(y_t^1 - x_t^1)\big) - \theta \end{bmatrix}. \tag{2.13}$$

The target needs to be tracked during a period T_{max} in a wooded area (Fig. 2.6), which affects the covariance of the measurement noise (see Fig. 2.5). The noise in the range measurement grows linearly with the distance between the sensor and the target but trees along the way make the growth faster. The bearing measurement noise increases linearly with the speed of the sensor. Thus, good range measurements require that the sensor is close to the target and not blocked by trees, while good bearing measurements require that the sensor moves slowly. The sensor has a maximum range of 15 meters, after which the noise covariance is infinite.

To employ the reduced value iteration algorithm (Alg. 2), the observation model (2.13) needs to be linearized about a predicted target trajectory during planning as described in Sec. 2.4.6. The Jacobian $\nabla_y h(x, y)$ of the observation model is presented in Appendix B.3 and B.4. As the linearization depends on the predicted target trajectory, it is necessary to re-plan the sensor trajectory online via model predictive control as shown in Alg. 3. The algorithm was implemented with $T_{max} = 100$, $T = 7$, $\epsilon = 0.1$, $\delta = 1$ and 100 Monte-Carlo simulations were carried out. Fig. 2.6 compares the tracking performance obtained via

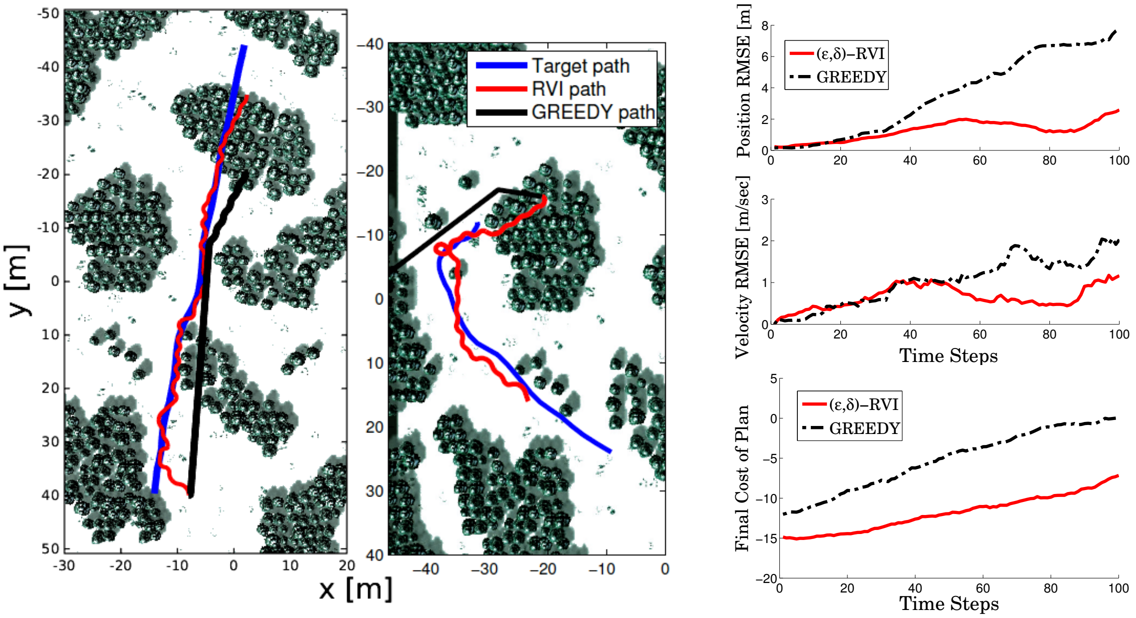


Figure 2.6: Simulation results from 100 Monte-Carlo runs of the target tracking scenario. Typical realizations are shown on the left. The average root-mean-square error (RMSE) of the estimated target position and velocity is shown on the right, along with the log det of the predicted target covariance on the bottom right.

reduced value iteration to that of the greedy policy (2.6). We can see that the greedy policy goes in straight line to keep the speed low, i.e., the bearing noise small, but cannot predict in advance that the range noise will increase as the target goes further away. As a result, the greedy solution is likely to lose the target.

2.5 Managing Complexity due to the Number of Sensors

So far we focused on managing the scalability of forward value iteration (Alg. 1) in T but neglected that the size $|\mathcal{U}| = |\mathcal{U}_1 \times \ldots \times \mathcal{U}_n|$ of the control set still grows exponentially with the number of sensors n. In this section, we develop a method to reduce the complexity in n to linear, which complements the (ϵ, δ)-reduced value iteration algorithm (Alg. 2) and leads to an effective and scalable approach for controlled information acquisition.

Before decentralizing the sensor control, we note that not all target states are of interest to all sensors. In particular, as we saw in Sec. 2.4.6, sometimes the state uncertainty of sensor i should be included in the target states and is of interest to sensor i only. To capture this, we decompose the target motion model in (2.10) further. Suppose that the dynamic part of the target state decomposes as $y_t^d := \begin{bmatrix}(y_{0,t}^d)^T & (y_{1,t}^d)^T & \cdots & (y_{n,t}^d)^T\end{bmatrix}^T$, where $y_{0,t}^d$ captures the evolution of an exogenous process, which the sensors need to track collaboratively, and $y_{i,t}^d$ captures the evolution of an endogenous process, only of interest to sensor i. The decomposition of the dynamics is:

$$\begin{aligned} y_{0,t+1}^d &= A_{0,t}^d y_{0,t}^d + w_{0,t}^d, & w_{0,t}^d &\sim \mathcal{N}(0, W_{0,t}^d), \\ y_{i,t+1}^d &= A_{i,t}^d(x_{i,t}, u_{i,t}) y_{i,t}^d + w_{i,t}^d, & w_{i,t}^d &\sim \mathcal{N}(0, W_{i,t}^d(x_{i,t}, u_{i,t})). \end{aligned} \tag{2.14}$$

The target states that are of interest to sensor i are $y_{i,t} := \begin{bmatrix}(y^d_{0,t})^T & (y^d_{i,t})^T & (y^s_t)^T\end{bmatrix}^T$.

Instead of solving the active information acquisition problem (2.2) jointly over all sensors, consider the following coordinate descent idea. Suppose that sensor 1 plans its own trajectory, individually, without taking the other sensors into account. In other words, it solves a single-sensor active information acquisition problem:

$$\mu^c_{1,0:(T-1)} \in \operatorname*{arg\,min}_{\hat{\mu}\in\mathcal{U}_1^T} J_T^{(1)}\left(\hat{\mu}\right).$$

Then, it passes its chosen control sequence (and if necessary the motion and observation models (2.1), (2.14) evaluated along it) on to sensor 2. Sensor 2 takes this information into account and plans its own trajectory by solving a two-sensor active information acquisition problem but with a fixed policy for sensor 1:

$$\mu^c_{2,0:(T-1)} \in \operatorname*{arg\,min}_{\hat{\mu}\in\mathcal{U}_2^T} J_T^{(2)}\left(\mu^c_{1,0:(T-1)}, \hat{\mu}\right).$$

Then, sensor 2 passes its information along to sensor 3, which solves a 3-sensor problem with fixed policies for the first 2 sensors. In general, sensor i needs the control sequences (and the motion and observation models (2.1), (2.14) evaluated along them), chosen by sensors 1 to $i-1$, and solves an i-sensor active information acquisition problem with fixed policies for the first $i-1$ sensors

$$\mu^c_{i,0:(T-1)} \in \operatorname*{arg\,min}_{\hat{\mu}\in\mathcal{U}_i^T} J_T^{(i)}\left(\mu^c_{1:(i-1),0:(T-1)}, \hat{\mu}\right). \tag{2.15}$$

Since the sensor labels are arbitrary, the coordinate descent can be carried out in any order. Because subproblem i in (2.15) has a search space of size $|\mathcal{U}_i|^T$, the coordinate descent scheme *reduces the complexity in n from exponential to linear*, i.e., from $O(|\mathcal{U}_1 \times \ldots \times \mathcal{U}_n|^T)$ to $O(\sum_{i=1}^n |\mathcal{U}_i|^T)$. Note that the approach is not fully-distributed because multihop communication is needed to pass the information but all-to-all communication is not required either because the optimization naturally occurs in communication hops. Each of the subproblems (2.15) can be solved via the (ϵ,δ)-RVI algorithm.

To gain intuition about the performance of the coordinate descent scheme with respect to the optimal centralized policy, we temporarily change the cost function in the active information acquisition problem (2.1) from $\sum_{t=1}^T \mathbf{h}(y_t \mid z_{1:t})$ to the more-commonly-used mutual information, $\mathbb{I}(y_{1:T}; z_{1:T})$.

Theorem 2.6. *Let the cost function in problem (2.1) be negative mutual information, $-\mathbb{I}(y_{1:T}; z_{1:T})$. Let μ^* be the optimal policy with associated cost J_T^*. Let μ^c be the coordinate descent policy with arbitrary sensor order and cost J_T^c. Suppose that the subproblems (2.15) are solved optimally. Then, the following guarantee holds for the performance of μ^c:*

$$J_T^c \geq J_T^* \geq 2J_T^c. \tag{2.16}$$

Since the cost is negative, this means that μ^c obtains at least 50% of the optimal value $-J^*$.

Remark: The proof relies on the fact that mutual information, when viewed as a function of the sensor set, is submodular. This result is different from (Singh et al. 2009a, Thm.1) because it applies to controlled dynamic Gaussian-Markov random fields and allows revisiting sensor states.

Although the coordinate descent has a theoretically appealing guarantee with respect to mutual information, Table 2.1 in Sec. 2.3 showed that when the target transitions are controlled (e.g., as in the active SLAM application in Sec. 2.5.1), mutual information might be inadequate as a value function. Next, we provide performance guarantees with respect to the additive conditional differential entropy cost function.

Corollary 2.7. *Consider problem* (2.1) *with additive conditional differential entropy,* $\sum_{t=1}^{T} \mathbf{h}(y_t \mid z_{1:t})$*, as cost function. Let* μ^* *be the optimal policy with associated cost* J_T^**. Let* μ^c *be a coordinate descent policy with arbitrary sensor order and cost* J_T^c*. Suppose that the subproblems* (2.15) *are solved optimally. Then:*

$$J_T^c \geq J_T^* \geq 2J_T^c - \sum_{t=1}^{T} \mathbf{h}(y_t) - \Delta_T,$$

where $\Delta_T := \sum_{i=1}^{n} \left(\sum_{t=1}^{T} \mathbf{h}(y_{1:i,t}) - \mathbf{h}(y_{1:i,1:T}) \right) \geq 0$*. If the dynamic target states are controlled (i.e.,* $A_{i,t}^d, W_{i,t}^d$ *depend on* $x_{i,t}, u_{i,t}$*), then* Δ_T *depends on* μ^**.*

2.5.1 Application: Multi-robot Active SLAM

To demonstrate the performance of our algorithms, we consider the multi-robot active SLAM problem. Trajectory optimization in SLAM is particularly challenging due to the coupling between the localization accuracy of the robots, the quality of the map, and the exploration of the environment. Leung et al. Leung et al. (2006) propose a nonmyopic approach for active SLAM with a single robot. They use attractors for exploration and an exhaustive search for trajectory optimization. Sim et al. Sim and Roy (2005) improve the efficiency by discretizing the environment and planning only for trajectories without self-intersections. There are also information-theoretic approaches which rely on non-Gaussian models (e.g. occupancy grids) but typically resort to greedy planning Bourgault et al. (2002), Tovar et al. (2006), Carlone et al. (2014). Recently, attention has been devoted to active visual SLAM as well Vidal-Calleja et al. (2006), Forster et al. (2014). When it comes to multi-robot active SLAM, there are very few approaches (Kontitsis et al. 2013, Pham and Juang 2013, Meyer et al. 2014) and most use greedy control.

We show that the active SLAM problem with n robots can be reduced to an instance of the active information acquisition problem (2.1) via linearization. Consequently, the combination of coordinate descent and (ϵ, δ)-reduced value iteration (Sec. 2.4.4) enable a decentralized nonmyopic solution that exploits sparsity. Since the focus is on the selection of informative controls, rather than the estimation aspect of the problem, we use an existing graph SLAM approach along the lines of Dellaert and Kaess (2006) for inference. In our simulations, the estimation is centralized but the control is decentralized. Decentralized estimation can be achieved as well, via Cunningham et al. (2013) or the techniques discussed in Sec. 2.6.

Let the robot states at time t be $r_{i,t}$ for $i = 1, \dots, n$ and let the dynamics of the i-th robot be:

$$r_{i,t+1} = f_i(r_{i,t}, u_{i,t}, \eta_{i,t}), \quad u_{i,t} \in \mathcal{U}_i, \tag{2.17}$$

where $\eta_{i,t}$ is zero-mean Gaussian noise with covariance E_i. The robots evolve in an environment with M landmarks with positions $m := [m_1^T, \dots, m_M^T]^T$. The task is to explore the environment, autonomously and efficiently, and create an accurate map of the landmark locations while staying well-localized. Each robot, depending on its pose $r_{i,t}$ can obtain measurements $z_{i,t}$ (typically nonlinear, e.g., range (Appendix B.3) and bearing (Appendix B.4)) of the visible landmarks according to the sensing model:

$$z_{i,t} = h_i(r_{i,t}, m) + v_{i,t}, \quad v_{i,t} \sim \mathcal{N}(0, V_i) \tag{2.18}$$

where $v_{i,t}$ is a Gaussian measurement noise, whose values are independent at any pair of times and across robots. We assume that the SLAM estimation layer provides a Gaussian prior on the robot poses r_0 and on the locations m_0 of the landmarks that have been *discovered so far*:

$$\begin{bmatrix} r_0 \\ m_0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \bar{r}_0 \\ \bar{m}_0 \end{bmatrix}, \begin{bmatrix} \Sigma_0^{rr} & \Sigma_0^{rm} \\ \Sigma_0^{mr} & \Sigma_0^{mm} \end{bmatrix} \right).$$

Problem (Multi-Robot Active SLAM)**.** *Given a planning horizon T, choose a sequence of functions $\mu_t(\mathcal{H}_t) \in \mathcal{U}$ for $t = 1, \ldots, T-1$, which optimizes:*

$$\begin{aligned} \min_{\mu_{0:(T-1)}} \; & \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}(r_t, m_t \mid z_{1:t}) \\ s.t. \quad & r_{t+1} = f(r_t, \mu_t(\mathcal{H}_t), \eta_t), \quad t = 0, \ldots, T-1, \\ & m_{t+1} = m_t, \qquad\qquad\qquad t = 0, \ldots, T-1, \\ & z_t = h(r_t, m_t) + v_t, \qquad\quad t = 1, \ldots, T, \end{aligned} \tag{2.19}$$

where $\mathcal{H}_0 := z_0$, $\mathcal{H}_t := (z_{0:t}, u_{0:(t-1)})$, $t > 0$ is the control and measurement history.

Because the robots neither know the *total number M* of landmarks, nor have prior information about the *locations of the undiscovered landmarks*, it is necessary to encourage them to explore the environment. We introduce dummy "exploration" landmarks with locations $l := [l_1^T, \ldots, l_{N_l}^T]^T$ at the current map frontiers (Yamauchi 1997) and specify a Gaussian prior on their locations with mean $\bar{l} := l$ and block diagonal covariance Σ^l with N_l blocks (see Fig. 2.7). This fake uncertainty in the exploration-landmark locations promises information gain to the robots. If it happens that there are no exploration landmarks within the reachable fields of view of the robots, the exploration process will stop because the algorithm we developed so far is unable to perceive information about the environment (even if known to the robots) beyond the planning horizon. Following Leung et al. (2006), we include attractor landmarks (see Fig. 2.7), which incorporate global information about the environment in the local planning process. For each robot we use a state machine as discussed in (Leung et al. 2006, Sec.V) to decide the attractor state from $\{$*None, Explore, Improve Map, Improve Localization*$\}$, with *None* having the lowest priority and *Improve Localization* - the highest. For instance, if the entropy of robot i's pose is larger than a threshold, we place localization attractors along the shortest path from the robot's estimated pose to the best localized landmark (see Fig. 2.7). We specify a Gaussian prior on the attractor locations $a := \begin{bmatrix} a_1^T & \cdots & a_{N_a}^T \end{bmatrix}^T$ with mean $\bar{a} := a$ and block diagonal covariance Σ^a with N_a blocks. To simplify notation, let $q_0 := \begin{bmatrix} m_0^T & l^T & a^T \end{bmatrix}^T$ be the combined locations of the discovered landmarks, the exploration landmarks, and the attractors

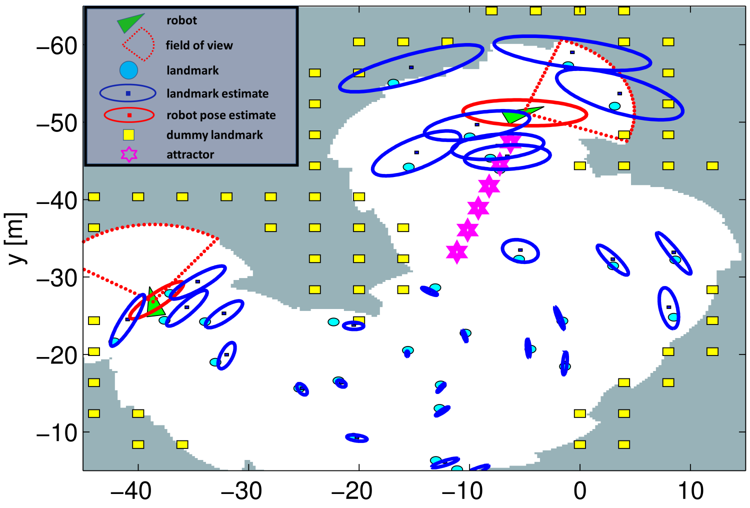


Figure 2.7: A snapshot of the multi-robot active SLAM simulation. The cyan circles show the true landmark positions, the blue squares and ellipses show the estimated landmark positions and covariances, the green triangles show the true robot poses, the red ellipses show the estimated robot position covariances, the dotted red sectors indicate the robots' fields of view, the yellow squares show the exploration landmarks, and the magenta stars show localization attractors associated with the top robot. The gray area represents unexplored space.

with prior:

$$\begin{bmatrix} r_0 \\ q_0 \end{bmatrix} = \begin{bmatrix} r_0 \\ m_0 \\ l \\ a \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \bar{r}_0 \\ \bar{m}_0 \\ \bar{l} \\ \bar{a} \end{bmatrix}, \begin{bmatrix} \Sigma_0^{rr} & \Sigma_0^{rm} & 0 & 0 \\ \Sigma_0^{mr} & \Sigma_0^{mm} & 0 & 0 \\ 0 & 0 & \Sigma^l & 0 \\ 0 & 0 & 0 & \Sigma^a \end{bmatrix} \right)$$

Finally, to reduce the active SLAM problem (2.19) to an instance of the active information acquisition problem (2.1), we use linearization and model predictive control as described in Sec. 2.4.6. The models in (2.19) are linearized about an open-loop predicted trajectory of the (deterministic) mean $(\bar{r}_t, \bar{q}_t)$ of the joint robot-landmark state (r_t, q_t). Let $\delta r_t := r_t - \bar{r}_t$, $\delta q_t := q_t - \bar{q}_t$, and $\delta z_t := z_t - h(\bar{r}_t, \bar{q}_t)$ be the deviations from the mean. We obtain the following linearized version of (2.19):

$$\begin{aligned} &\min_{\mu_{0:(T-1)}} \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}(r_t, q_t \mid z_{1:t}) \\ &\text{s.t.} \quad \text{for } t = 0, \dots, T-1 \\ &\bar{r}_{t+1} = f(\bar{r}_t, \mu_t(\mathcal{I}_t), 0), \; \bar{q}_{t+1} = \bar{q}_t, \; \delta q_{t+1} = \delta q_t \\ &\delta r_{t+1} \approx \left[\frac{\partial f}{\partial r}(\bar{r}_t, \mu_t(\mathcal{I}_t), 0)\right] \delta r_t + \left[\frac{\partial f}{\partial \eta}(\bar{r}_t, \mu_t(\mathcal{I}_t), 0)\right] \eta_t \\ &\delta z_{t+1} \approx \left[\frac{\partial h}{\partial r}(\bar{r}_{t+1}, \bar{q}_{t+1})\right] \delta r_{t+1} + \left[\frac{\partial h}{\partial m}(\bar{r}_{t+1}, \bar{q}_{t+1})\right] \delta q_{t+1} + v_t, \end{aligned} \tag{2.20}$$

which is an instance of problem (2.1) with target state $y_t := (\delta r_t^T \; \delta q_t^T)^T$ and sensor state $x_t := (\bar{r}_t^T \; \bar{q}_t^T)^T$. Our solution to the active SLAM problem is summarized in Alg. 4 and is

Algorithm 4 Multi-Robot Active SLAM

```
repeat: {
    1) Receive measurements and update the SLAM estimate
    2) Remove any exploration landmarks within the robots'
       fields of view and add new ones at the map frontiers
    3) Remove the old attractors and add new ones if necessary
    4) Plan $T$-step trajectories by solving (2.20) via coordinate
       descent (2.15) and reduced value iteration (Alg. 2)
    5) Apply the first control inputs to move each robot
}
```

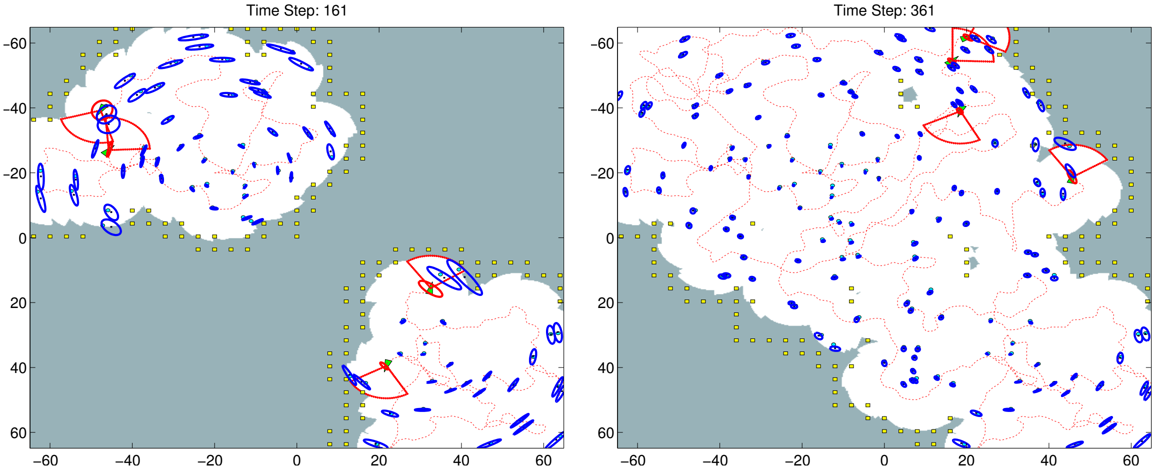


Figure 2.8: Two instances of a four-robot active SLAM simulation which demonstrate the estimation quality and the exploration progress in an environment with 200 landmarks. The red dotted curves show the estimated robot trajectories, while the other symbols are described in the caption of Fig. 2.7. The robots have differential drive dynamics with maximum velocity 3 m/s, standard deviations 0.1 m/s and $5°/s$ in linear and angular velocities, respectively, 10 m sensing ranges, 94° fields of view, and used range and bearing measurements with standard deviations 0.15 m and 5°, respectively. See Appendix D.16 Extension 1 for a video of the simulation.

illustrated in Fig. 2.8. Cooperation among the robots is achieved via the coordinate descent scheme, which allows the robots to take information from their teammates into account during the planning process. Details about the robot motion and measurement models are presented in the caption of Fig. 2.8. The performance is quantified in Fig. 2.9. Importantly, the entropy in the landmark positions is decreasing over time, although new landmarks are continuously being discovered. The robot pose entropies fluctuate because the robots need to repeatedly sacrifice localization accuracy in order to explore the environment. The plots indicate that the robots successfully explore the environment, create an accurate map of the landmark positions, and remain localized well in the process. The performance with a 12-step planning horizon is, as expected, better than greedy planning. However, the message here is not that our algorithm is better than a greedy approach, but that, if time for computation is available, it should be used to improve the planned trajectories. The (ϵ, δ) parameters of the reduced value iteration algorithm allow us to utilize the computation time effectively and the extra work is guaranteed to improve the performance, compared to the greedy approach baseline.

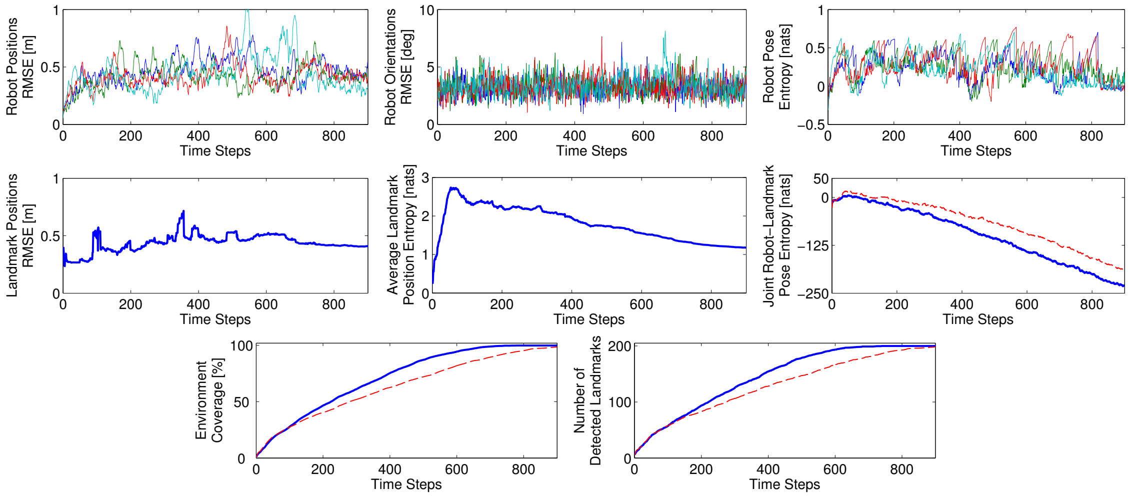


Figure 2.9: The top three plots show the root mean square error (RMSE) in the robot position estimates, the RMSE in the robot orientation estimates, and the entropy of the robot pose estimates, average over 15 repetitions of the four-robot active SLAM scenario in Fig. 2.8. The middle row shows the RMSE in the landmark position estimates, the average entropy in the landmark position estimates, and the entropy of the joint robot-landmark-pose estimate. The last two plots show the percentage of the environment covered by the robots and the number of detected landmarks over time. The robot trajectories were planned using RVI (Alg. 2) with planning horizon $T = 12$, $\epsilon = \infty$, and $\delta = 1.5$. The red dotted curves on the last three plots show the performance when the trajectories are obtained via a greedy policy (RVI with $T = 1$).

2.6 Distributed Estimation

This chapter discussed information-seeking control and how to manage the computational complexity in the planning horizon T via reduced value iteration (Sec. 2.4.4) and in the number of sensors n via decentralized control (Sec. 2.5). Due to the linear Gaussian assumptions on the motion and observation models, the target inference was handled by the Kalman filter. In many applications, however, decentralized control goes hand in hand with decentralized target inference. In the final section of this chapter, we focus solely on the estimation aspect and discuss algorithms for distributed target tracking and sensor self-localization.

Consider a team of sensors with states (informally, locations) $\{x_1, \ldots, x_n\} \subset \mathcal{X} \cong \mathbb{R}^{d_x}$. Assume that the sensors are static and connected by a communication network, represented by an undirected graph $G = (\{1, \ldots, n\}, E)$ with vertices corresponding to the sensors and $|E| = m$ edges. An edge $(j, i) \in E$ from sensor j to sensor i exists if they can communicate. The set of nodes (neighbors) connected to sensor i is denoted by N_i. As before, the task of the sensors is to estimate and track the state $y_t \in \mathcal{Y} \cong \mathbb{R}^{d_y}$ of mobile target(s), where $\mathcal{Y}$ is a convex set. The target evolves according to the motion model:

$$y_{t+1} = Ay_t + w_t, \qquad w_t \sim \mathcal{N}(0, W). \tag{2.21}$$

Sensor i, depending on its location x_i, obtains measurements $z_{i,t}$ of the target state according to the model:

$$z_{i,t} = H_i(x_i)y_t + v_{i,t}, \;\; v_{i,t} \sim \mathcal{N}(0, V_i(x_i)). \tag{2.22}$$

As before, the measurement noise is independent at any pair of times and across different sensors. It is also independent of the target process noise w_t. Individual measurements, $z_{i,t}$, although potentially informative, do not reveal the target state completely, i.e. each sensor faces a local identification problem. We assume, however, that the target is observable if one has access to the measurements received by all sensors.

To use the measurements $z_{i,t}$ for target inference, the sensors need to know their locations. However, it is common, especially in large networks, that they have only a rough estimate (prior). We suppose that each sensor has access to relative measurements of the locations of its neighbors, which can be used to localize it. In particular, at time t sensor i receives the following noisy relative state measurement from its neighbor j:

$$s_t^{ij} = x_j - x_i + \epsilon_t^{ij}, \qquad \epsilon_t^{ij} \sim \mathcal{N}(0, \mathcal{E}_{ij}), \tag{2.23}$$

where ϵ_t^{ij} is a measurement noise, which is independent at any pair of times and across sensor pairs. The relative measurement noises are independent of the target measurement and motion noises too. Since there is translation ambiguity in the measurements (2.23) we assume that all sensors agree to localize themselves in the reference frame of sensor 1. The location estimates can then be used in place of the unknown sensor positions during the target estimation procedure. The joint localization and estimation problem is summarized below.

Problem (Joint Estimation and Localization)**.** *The task of each sensor i is to construct estimators $\hat{x}_{i,t}$ and $\hat{y}_{i,t}$ of its own location x_i and of the target state y_t in a distributed manner, i.e., using information only from its neighbors and the measurements $\{s_t^{ij} \mid j \in \mathcal{N}_i\}$ and $\{z_{i,t}\}$.*

2.6.1 Distributed Target Tracking

We begin with the task of distributed target tracking, assuming that the sensors know their locations. We specialize the general parameter estimation scheme of Rahnama Rad and Tahbaz-Salehi (2010) to linear Gaussian observation models such as (2.22). We show that the resulting distributed linear filter is mean-square consistent[3] when the target is stationary. This result is stronger than the weak consistency[3] shown in the general non-Gaussian case in Rahnama Rad and Tahbaz-Salehi 2010, Thm. 1.

To introduce the estimation scheme from Rahnama Rad and Tahbaz-Salehi (2010), suppose that the target is static, i.e., $y := y_0 = y_1 = \dots$ and that, instead of the linear Gaussian measurements in (2.22), the sensor receive measurements, $z_{i,t}$, drawn from a general distribution with conditional probability density function $l_i(\cdot|y)$. As before, the signals observed by sensor i are i.i.d. over time and independent from the observations of all other sensors. In order to aggregate the information provided to it, each sensor i holds and updates a pdf $p_{i,t}$ over the target state space $\mathcal{Y}$. Consider the following distributed

[3] A distributed estimator of a parameter y is *weakly consistent* if all estimates, $\hat{y}_{i,t}$, converge in probability to y, i.e., $\lim_{t\to\infty} \mathbb{P}\big(\|\hat{y}_{i,t} - y\| \geq \epsilon\big) = 0$ for any $\epsilon > 0$ and all i. It is *mean-square consistent* if all estimates converge in L^2 to y, i.e., $\lim_{t\to\infty} \mathbb{E}\big[\|\hat{y}_{i,t} - y\|^2\big] = 0, \forall i$.

estimation algorithm:

$$\begin{aligned} p_{i,t+1}(y) &= \xi_{i,t} l_i(z_{i,t+1} \mid y) \prod_{j \in \mathsf{N}_i \cup \{i\}} \big(p_{j,t}(y)\big)^{\kappa_{ij}} \\ \hat{y}_{i,t} &\in \arg\max_{y \in \mathcal{Y}} p_{i,t}(y) \end{aligned} \tag{2.24}$$

where $\xi_{i,t}$ is a normalization constant ensuring that $p_{i,t+1}$ is a proper pdf and $\kappa_{ij} > 0$ are weights such that $\sum_{j \in \mathsf{N}_i \cup \{i\}} \kappa_{ij} = 1$. The update is the same as the standard Bayes rule except that sensor i does not just use its own prior but a *geometric average* of its neighbors' priors. Given a connected graph, Rahnama Rad and Tahbaz-Salehi (2010) show that (2.24) is weakly consistent under broad assumptions on the observation models l_i.

Next, we specialize the estimator in (2.24) to the linear Gaussian measurement model in (2.22). Let $\mathcal{G}(\omega, \Omega)$ denote a Gaussian distribution (in information space) with mean $\Omega^{-1}\omega$ and covariance matrix Ω^{-1}. The quantities ω and Ω are conventionally called *information vector* and *information matrix*, respectively. Suppose that the pdfs $p_{i,t}$ of all sensors $i \in \{1, \dots, n\}$ at time t are that of Gaussian distributions $\mathcal{G}(\omega_{i,t}, \Omega_{i,t})$. We claim that the posteriors resulting from applying the update in (2.24) remain Gaussian.

Lemma 2.8 (Barker et al. 1995, Thm. 2)**.** *Let $Y_i \sim \mathcal{G}(\omega_i, \Omega_i)$ for $i = 1, \dots, n$ be a collection of random Gaussian vectors with associated weights κ_i. The weighted geometric mean, $\prod_{i=1}^n p_i^{\kappa_i}$, of their pdfs p_i is proportional to the pdf of a random vector with distribution $\mathcal{G}\left(\sum_{i=1}^n \kappa_i \omega_i, \sum_{i=1}^n \kappa_i \Omega_i\right)$.*

Lemma 2.9 (Barker et al. 1995, Thm. 2)**.** *Let $Y \sim \mathcal{G}(\omega, \Omega)$ and $\mathcal{V} \sim \mathcal{G}(0, V^{-1})$ be random vectors. Consider the linear transformation $Z = HY + \mathcal{V}$. The conditional distribution of $Y \mid Z = z$ is proportional to $\mathcal{G}(\omega + H^T V^{-1} z, \Omega + H^T V^{-1} H)$.*

Lemma 2.8 states that if the target priors are Gaussian, then after the geometric averaging in (2.24), the average distribution will also be Gaussian and its information vector and information matrix will be weighted averages of the prior ones. Lemma 2.9 says that after applying Bayes rule the distribution remains Gaussian. Combining the two allows us to derive the following linear Gaussian version of the estimator in (2.24):

$$\begin{aligned} \omega_{i,t+1} &= \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \omega_{j,t} + H_i^T V_i^{-1} z_{i,t}, \\ \Omega_{i,t+1} &= \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \Omega_{j,t} + H_i^T V_i^{-1} H_i, \\ \hat{y}_{i,t} &:= \Omega_{i,t}^{-1} \omega_{i,t}, \end{aligned} \tag{2.25}$$

where $H_i := H_i(x_i)$ and $V_i := V_i(x_i)$. We prove a strong result about the quality of the estimates in this linear Gaussian case.

Theorem 2.10. *Suppose that the communication graph G is connected and the matrix $\begin{bmatrix} H_1^T & \cdots & H_n^T \end{bmatrix}^T$ has rank d_y. Then, the estimates in (2.25) of all sensors converge in mean square to y, i.e., $\lim_{t \to \infty} \mathbb{E}\big[\|\hat{y}_{i,t} - y\|_2^2\big] = 0$ for all i.*

Algorithm 5 Distributed Linear Estimator

```
Input: Prior $(\omega_{i,t}, \Omega_{i,t})$, messages $(\omega_{j,t}, \Omega_{j,t}), \forall j \in \mathcal{N}_i$, and measurement $z_{i,t}$
Output: $(\omega_{i,t+1}, \Omega_{i,t+1})$

Update Step:          $\omega_{i,t+1} = \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \omega_{j,t} + H_i^T V_i^{-1} z_{i,t}$
                      $\Omega_{i,t+1} = \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \Omega_{j,t} + H_i^T V_i^{-1} H_i$
                      $\hat{y}_{i,t+1} = \Omega_{i,t+1}^{-1} \omega_{i,t+1}$
Prediction Step:      $\Omega_{i,t+1} = (A \Omega_{i,t+1}^{-1} A^T + W)^{-1}$
                      $\omega_{i,t+1} = \Omega_{i,t+1} A \hat{y}_{i,t+1}$
```

The procedure in (2.25) can be extended to track dynamic targets as in (2.21) by adding a local prediction step, same as that of the Kalman filter, at each sensor. The resulting distributed linear filter is summarized in Alg. 5 and Thm. 2.10 guarantees its mean square consistency for static targets. Its performance on dynamic targets is studied in a vehicle tracking scenario in Sec. 2.6.4.

2.6.2 Node Localization from Relative Measurements

Target tracking via the distributed estimator in Alg. 5 requires that the true sensor locations are known. As mentioned earlier this is typically not the case, especially for large sensor networks. This section describes a method for localization from relative measurements (2.23), whose strong convergence guarantees can be used to analyze the convergence of a joint localization and estimation procedure. The relative measurements, received by all sensors at time t, can be written in matrix form as follows:

$$s_t = (B \otimes I_{d_x})^T x + \epsilon_t,$$

where $B \in \mathbb{R}^{n \times m}$ is the incidence matrix of the communication graph G. All sensors agree to localize relative to node 1 and know that $x_1 = 0$. Let $\tilde{B} \in \mathbb{R}^{(n-1) \times m}$ be the incidence matrix with the row corresponding to sensor 1 removed. Further, define $\mathcal{E} := \mathbb{E}[\epsilon_t \epsilon_t^T] = \mathbf{diag}(\mathcal{E}_1, \ldots, \mathcal{E}_m)$, where $\{\mathcal{E}_k\}$ is an enumeration of the noise covariances associated with the edges of G. Given t measurements, the least squares estimate of x leads to the classical best linear unbiased estimator:

$$\hat{x}_t := \left(\tilde{B} \mathcal{E}^{-1} \tilde{B}^T\right)^{-1} \tilde{B} \mathcal{E}^{-1} \sum_{\tau=0}^{t-1} s_\tau, \tag{2.26}$$

where the inverse of $\tilde{B} \mathcal{E}^{-1} \tilde{B}^T$ exists as long as the graph G is connected (Barooah and Hespanha 2007). The computation in (2.26) can be distributed via a Jacobi algorithm for solving a linear system as follows. At time t, each sensor maintains an estimate $\hat{x}_{i,t}$ of its own state and a history of the averaged measurements, $\sigma_{i,t} := \frac{1}{t+1} \sum_{\tau=0}^{t} \sum_{j \in \mathsf{N}_i} \mathcal{E}_{ij}^{-1} s_\tau^{ij}$, received up to time t. Given prior estimates $(\hat{x}_{i,t}, \sigma_{i,t})$, the update of the distributed Jacobi

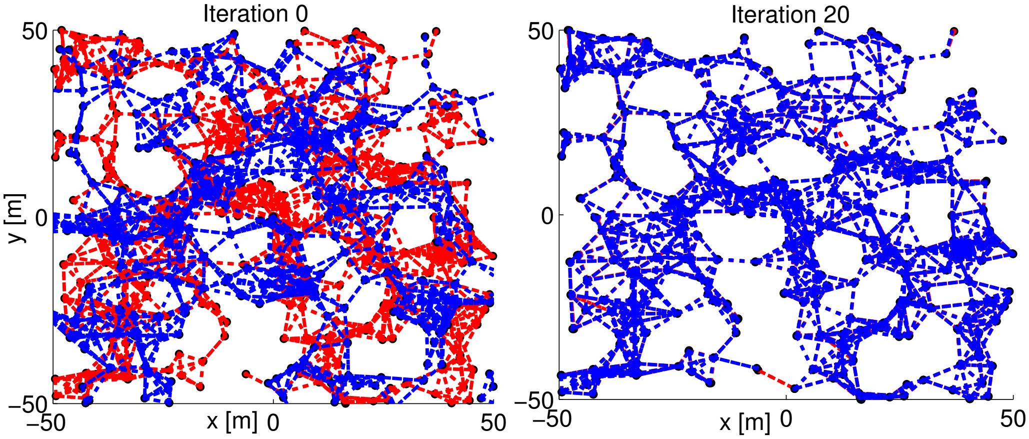


Figure 2.10: Initial and final (after 20 steps) node locations (red) estimated by the distributed localization algorithm on a randomly generated graph with 300 nodes (blue) and 1288 edges (blue dotted lines)

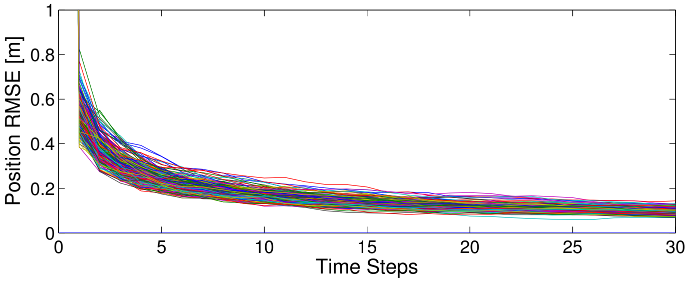


Figure 2.11: Root mean squared error of the location estimates obtained from averaging 50 simulated runs of the distributed localization algorithm with randomly generated graphs with 300 nodes (e.g., Fig. 2.10), connectivity radius 10 m, and measurement covariances $\mathcal{E}_{ij} = I_2$

algorithm at sensor i is:

$$\begin{aligned} \hat{x}_{i,t+1} &= \Big(\sum_{j\in\mathcal{N}_i} \mathcal{E}_{ij}^{-1}\Big)^{-1}\Big(\sum_{j\in\mathcal{N}_i} \mathcal{E}_{ij}^{-1}\hat{x}_{j,t} - \sigma_{i,t}\Big), \\ \sigma_{i,t+1} &= \frac{1}{t+1}\Big(t\sigma_{i,t} + \sum_{j\in\mathcal{N}_i} \mathcal{E}_{ij}^{-1} s_t^{ij}\Big). \end{aligned} \tag{2.27}$$

Barooah and Hespanha (2007), Barooah (2007) show that, with a *single round* of measurements, the Jacobi algorithm provides an unbiased estimate of x. Here, we incorporate sequential measurements and prove a stronger performance guarantee.

Theorem 2.11. *Suppose that the communication graph G is connected. Then, the estimates $\hat{x}_{i,t}$ of the sensor states in (2.27) are mean-square and strongly consistent estimators of the true sensor states, i.e., for all i:*

$$\lim_{t\to\infty} \mathbb{E}\left[\|\hat{x}_{i,t} - x_i\|_2^2\right] = 0, \quad \mathbb{P}\left(\lim_{t\to\infty} \|\hat{x}_{i,t} - x_i\|_2 = 0\right) = 1$$

The performance of the distributed localization algorithm was analyzed on randomly generated graphs. An instance of the localization task is illustrated in Fig. 2.10, while the estimation error is shown in Fig. 2.11. The location priors were chosen from a normal distribution with standard deviation of 5 meters from the true node positions.

2.6.3 Joint Localization and Estimation

Having derived separate estimators for the sensor locations and the target state, we are ready to return to the original problem of joint localization and estimation. At time t, the location estimates $\{\hat{x}_{i,t}\}$ in (2.27) can be used in the target estimator (2.25) instead of the true sensor positions. It is important to analyze the evolution of the coupled procedure because it is not clear that the convergence result in Thm. 2.10 will continue to hold. A regularity assumption which stipulates that nearby sensing locations provide similar information gain is necessary.

Assumption. *The sensor matrices*[4] $M_i(x) := H_i(x)^T V_i(x)^{-1} H_i(x)$ *are bounded continuous functions of* x *for all* i*.*

The following theorem ensures that the target state estimator retains its convergence properties when used jointly with the distributed localization procedure.

Theorem 2.12. *Let* $\{\hat{x}_{i,t}\}$ *be strongly consistent estimators of the sensor states, i.e.,* $\hat{x}_{i,t} \xrightarrow{a.s.} x_i, \forall i$*. Suppose that the communication graph* G *is connected and the matrix* $\begin{bmatrix} H_1(x_1)^T & \cdots & H_n(x_n)^T \end{bmatrix}^T$ *has rank* d_y*. Let* $\delta > 0$ *be arbitrary. If each sensor* i *updates its target estimate* $(\omega_{i,t}, \Omega_{i,t})$ *as follows:*

$$\begin{aligned} \omega_{i,t+1} &= \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \omega_{j,t} + \widehat{H}_{i,t}^T \widehat{V}_{i,t}^{-1} z_{i,t}, \\ \Omega_{i,t+1} &= \sum_{j \in \mathcal{N}_i \cup \{i\}} \kappa_{ij} \Omega_{j,t} + \widehat{H}_{i,t}^T \widehat{V}_{it}^{-1} \widehat{H}_{i,t}, \\ \hat{y}_{i,t+1} &= \left(\Omega_{i,t+1} + (t+1)\delta I_{d_x}\right)^{-1} \omega_{i,t+1}, \end{aligned} \tag{2.28}$$

where $\widehat{H}_{i,t} := H_i(\hat{x}_{i,t})$ *and* $\widehat{V}_{i,t} := V_i(\hat{x}_{i,t})$*, then the asymptotic mean square error of target estimates is* $O(\delta^2)$*:*

$$\lim_{t \to \infty} \mathbb{E}\left[\|\hat{y}_{i,t} - y\|_2^2\right] = \delta^2 y^T \left(\sum_{j=1}^n \pi_j M_j(x_j) + \delta I\right)^{-2} y, \quad \textit{for all } i,$$

where y *is the true target state,* x_j *is the true position of sensor* j*, and* $\pi \in \mathbb{R}^n$ *is the unique stationary distribution of the aperiodic irreducible Markov chain with transition matrix* $\mathcal{K} := [\kappa_{ij}]$*.*

According to Thm. 2.12, the combined procedure for estimating the sensor locations and the target state, specified by (2.27) and (2.28), has an arbitrarily small mean square error.

[4] It is natural to describe sensor properties in terms of the sensor matrix because, in an analogy with the Kalman filter, it captures the amount of information added to the information matrix during an update step of the Riccati map.

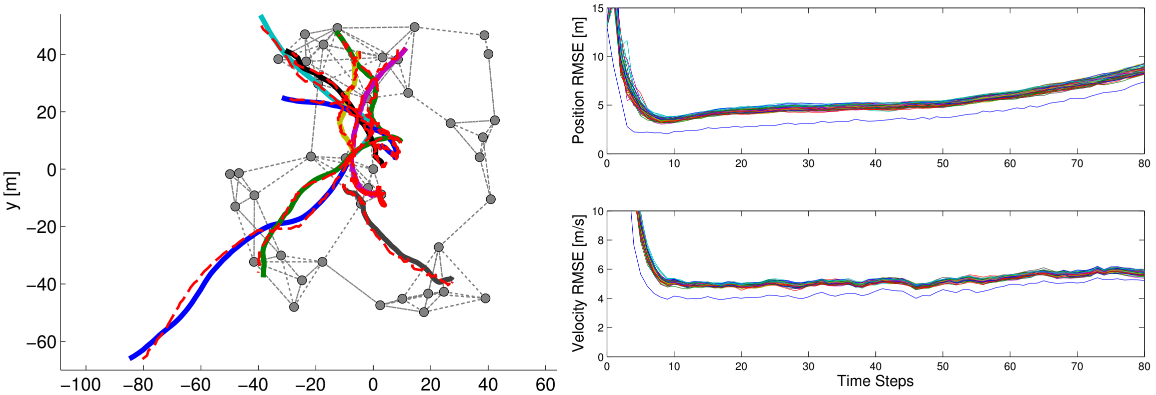


Figure 2.12: The left plot shows a realization of the vehicle tracking scenario in which a sensor network with 40 nodes (grey) tracks 10 mobile vehicles via range and bearing measurements. The true and estimated vehicle trajectories are shown by solid curves and red dotted curves, respectively. The root mean squared error of the estimated vehicle positions and velocities obtained from averaging 50 simulated runs of the distributed linear estimator are shown on the right. The error increases because as the vehicles move away from the sensor network the covariance of the measurement noise grows with the distance ($V(d_{ij}) = \mathbf{diag}((0.07d_{ij} + 0.04)^2 \text{ m}^2, (d_{ij} + 2)^2 \text{ deg}^2)$ for $d_{ij} := \|x_i - y_j\|_2$). The errors of node 1 (blue) are lower because its location is known ($x_1 = 0$) and always close to the starting vehicle positions. The rest of the parameters were: $n = 40, q = 1.5, \tau = 0.2, \mathcal{E}_{ij} = 0.5I_2$.

2.6.4 Application: Mobile Vehicle Tracking via a Sensor Network

First, we study the performance of the distributed linear filter (Alg. 5) in a vehicle tracking scenario. We consider tracking several mobile vehicles via a sensor network using range and bearing measurements (see Fig. 2.12). Similar to Sec. 2.4.8, the position $(y_j^1, y_j^2) \in \mathbb{R}^2$ and velocity $(\bar{y}_j^1, \bar{y}_j^2) \in \mathbb{R}^2$ of the j-th target have discretized double-integrator dynamics driven by Gaussian noise:

$$y_{j,t+1} = \begin{bmatrix} I_2 & \tau I_2 \\ 0 & I_2 \end{bmatrix} y_{j,t} + \eta_{j,t}, \quad W := q \begin{bmatrix} \frac{\tau^3}{3} I_2 & \frac{\tau^2}{2} I_2 \\ \frac{\tau^2}{2} I_2 & \tau I_2 \end{bmatrix},$$

where $y_j = [y_j^1, y_j^2, \bar{y}_j^1, \bar{y}_j^2]^T$ is the j-th target state, τ is the sampling period is *sec*, and q is a diffusion strength measured in $(\frac{m}{sec^2})^2 \frac{1}{Hz}$. Each sensor in the network takes noisy range and bearing measurements of each target's position:

$$z_{i,j,t} = \begin{bmatrix} \sqrt{(y_j^1 - x_i^1)^2 + (y_j^2 - x_i^2)^2} \\ \arctan\big((y_j^2 - x_i^2)/(y_j^1 - x_i^1)\big) \end{bmatrix} + v(t, x_i, y_j),$$

where $x_i := (x_i^1, x_i^2) \in \mathbb{R}^2$ is the sensor's location and the noise v grows linearly with the distance between the sensor and the target. The observation model is nonlinear in this case so we resort to linearization with respect to y_j in order to apply the distributed linear filter (Alg. 5). The tracking error is presented in Fig. 2.12.

2.6.5 Application: Methane Emission Monitoring via a Sensor Network

The performance of the joint procedure for estimating the sensor locations and the target state, specified by (2.27) and (2.28), was evaluated on the methane emission monitoring problem introduced in Sec. 2.4.7. In this case, a static sensor network of remote methane leak detectors (see Fig. 2.3 for details) is deployed to estimate the methane concentration in the landfill. As before, the sensor observation model is linear and fits the form in (2.22). Also, the methane field is assumed static ($A = I_{d_y}, W = 0$) and is modeled by discretizing the environment into cells and representing the gas concentration with a Gaussian random field, $y \in \mathbb{R}^{d_y}$. The estimation results are summarized in Fig. 2.13 and show that the continuity assumption on the sensor matrices in Sec. 2.6.3 is important.

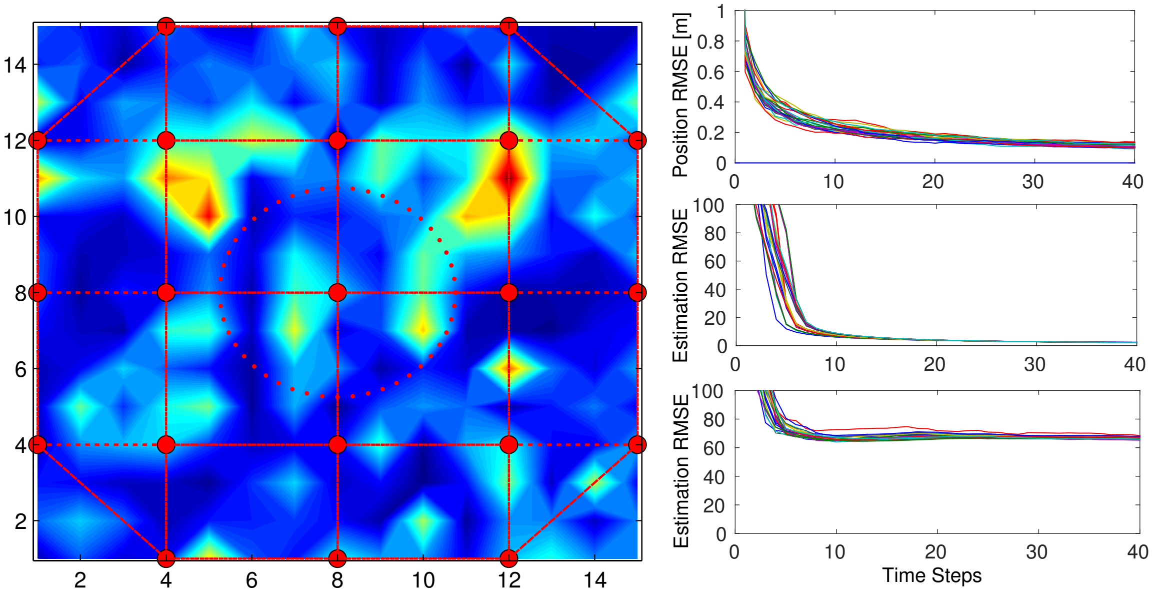


Figure 2.13: Methane emission monitoring via a sensor network. The true (unknown) sensor locations (red nodes), the sensing range (red circle), and a typical realization of the methane field are shown on the left. The gas concentration varies from 0 to 800 parts per million (ppm) and the standard deviation of the measurements is 5 ppm. The root mean squared error of the location estimates and of the field estimates obtained from averaging 50 simulated runs of the joint localization and estimation algorithm with continuous sensor observation models are shown in the top two plots on the right (for $\mathcal{E}_{ij} = I_2$). In an additional experiment, the sensors were placed on the boundaries of the cells of the discretized field. As the observation model for each sensor was defined in terms of the proximal environment cells, this made the observation models discontinuous. The bottom right plot illustrates that the concentration estimation error does not vanish when discontinuities are present.

2.7 Summary

This chapter considered active information acquisition under the assumption that the sensor observation models and the target motion models are linear in the target state and perturbed by Gaussian noise. We discussed that conditional entropy is an appropriate measure of uncertainty and proved that the classical separation principle between estimation and control holds for the linear Gaussian information acquisition problem. As a result, we managed to reduce the original stochastic optimal control problem to a deterministic version but still the complexity of the optimal centralized algorithm turned out to be exponential both in the length of the planning horizon and in the number of sensors. First, we proposed a reduced value iteration algorithm that mitigates the complexity in the planning horizon by discarding uninformative sensor trajectories from the search space and exploits sparsity in the target information matrix. We proved that the suboptimality of the algorithm is bounded regardless of the length of the planning horizon. Second, we developed a decentralized control strategy that achieves linear complexity in the number of sensors and offers performance guarantees with respect to the optimal centralized algorithm. Finally, we focused on distributed target inference and sensor self-localization. We developed a distributed Kalman filter for tracking mobile targets and a distributed Jacobi algorithm for sensor localization and proved that the combined localization and target estimation procedure has arbitrarily small asymptotic mean square error. We presented applications in methane emission monitoring, mobile vehicle tracking, and active exploration, localization, and mapping with multiple robots. The techniques developed in this chapter offer a scalable nonmyopic approach for active information gathering and, most importantly, coupled with linearization and model predictive control, they can generate closed-loop control policies for a broad variety of practical applications with nonlinear observation and motion models.

Chapter 3

Greedy Information Acquisition with Unknown Data Association or Unknown Observation Models

Ch. 2 presented one approach for approximately solving the general active information acquisition problem. The approach uses statistical linearization to convert the original problem into a linear Gaussian one and model predictive control to incorporate information about the measurement realizations into the otherwise open-loop plan. Consequently, it provides good results when the motion and sensing models are accurate and the uncertainties are unimodal. However, the framework requires external handling of sensing nuisances such as missed detections, false alarms, or unknown data association (i.e., the assignment of visible targets to received measurements) and is not applicable at all in cases when the models cannot be linearized (e.g., the state or measurement spaces are discrete) or are completely unknown. This chapter considers two specific applications of practical importance in which the aforementioned complications arise. A more general view of active information acquisition in discrete state and measurement spaces is presented in the next chapter.

In the first part of this chapter, we revisit the active localization problem from Sec. 2.5.1, in which a sensor plans informative actions in order to improve the uncertainty in its own location, but this time we add a twist. Instead of using low-level geometric features such as points, lines, and planes, we recognize objects in the sensor's surroundings and consider localizing it within a prior map of semantically-meaningful landmarks. As object classes are discrete observations and the association between detected and visible objects is unknown, we cannot use the Kalman filter for inference and hence cannot apply the techniques from Ch. 2. Instead of the traditional vector-based representation, we propose a sensor observation model that encodes the (discrete) semantic observations via random finite sets and enables a unified treatment of missed detections, false alarms, and data association. The advantage of localizing against semantically-meaningful landmarks is that it is less ambiguous and can be used for global localization[1] and loop-closure[2]. Also, semantically-annotated

[1] Visual-odometry and SLAM techniques typically do not use an absolute world reference and do not provide global localization. The initial robot pose is chosen as the map origin and is tracked over time.

[2] Loop-closure is the process of detecting that a mobile sensor is at a previously visited location and updating the localization (and map in the case of SLAM) uncertainty accordingly.

maps can be constructed for GPS-denied environments via the mapping approaches that received significant attention in recent years (Kostavelis and Gasteratos 2013, Galindo et al. 2005, Pronobis 2011, Nüchter and Hertzberg 2008, Civera et al. 2011).

Existing landmark-based localization and SLAM techniques require external solutions to the problems of data association and clutter rejection (Bailey 2002, Montemerlo and Thrun 2003). Moreover, state-of-the-art approaches nowadays are based on factor graphs (Kaess et al. 2008, Kummerle et al. 2011) and rely heavily on continuous Gaussian random variables. Hence, they cannot handle (discrete non-Gaussian) object labels in the estimation. There is a line of work addressing visual localization, which matches observed image features to an image database, whose images correspond to the nodes of a topological map (Wolf et al. 2005, Se et al. 2005, Angeli et al. 2009, Wang et al. 2006, Mariottini and Roumeliotis 2011, Košecká and Li 2004). Wang et al. (2006) represent each location in the topological map by a set of interest points that can be reliably detected in images and use nearest neighbor search to match observed SIFT features to the database. Košecká and Li (2004) also characterize scale-invariant key points by the SIFT descriptor and find nodes in the topological map, whose features match the observed ones the best. The drawback of this most likely data association approach is that when it is wrong it quickly causes the estimation procedure to diverge. Hesch et al. (2013) study the effects of unobservable directions on the estimator consistency in vision-aided inertial navigation systems. In the SLAM context, bad data association can be mitigated by a two-stage approach, in which the back-end optimizer (e.g., factor graph) is allowed to reject or alter associations proposed by the front-end (e.g., appearance-based place recognition) (Sünderhauf and Protzel 2011). As object recognition algorithms miss detections and produce false alarms, correct data association is crucial for semantic localization and semantic world modeling (Wong et al. 2013).

In recent years, random-finite-set-based solutions to SLAM gained popularity due to their unified treatment of filtering and data association. Mahler (2007) derived the Bayesian recursion with random-finite-set-valued observations and proposed a first-moment approximation, called the probability hypothesis density (PHD) filter. The PHD filter has been successfully applied to SLAM by Kalyan et al. (2010), Lee et al. (2013), and Mullane et al. (2011). In these works, the vehicle trajectory is tracked by a particle filter and the first moment of a trajectory-conditioned map for each particle is propagated via a Gaussian-mixture PHD filter. Bishop and Jensfelt (2010) address global geometric localization by formulating hypotheses about the robot state and tracking them with the PHD filter. Zhang et al. (2012) propose an approach for visual odometry using a PHD filter to track SIFT features extracted from observed images. Most of the random-set approaches rely on a first-moment approximation via the PHD filter. Only few deal with the full observation model (Dames et al. 2013, Ma et al. 2006, Sidenbladh and Wirkander 2003) and none have applied the model in a semantic setting or studied its computational complexity. There are also several semantic localization approaches that do not rely on random finite sets. Anati et al. (2012) match histogram-of-gradient-energies and quantized-colors features to the expected features from a prior semantic map. Yi et al. (2009) and Ko et al. (2013) use semantic descriptions of distance and bearing in a contextual map for active semantic localization. Bao and Savarese (2011) propose a maximum-likelihood-estimation formulation for semantic structure from motion. In addition to recovering camera parameters (motion) and 3-D locations of image features (structure), the authors recover the 3-D locations, orientations, and categories of objects in the scene. A Markov-chain-Monte-Carlo algorithm is used to solve a batch estimation problem by sampling from the likelihood of the collected measurements.

In the second part of the chapter, we look at an even more extreme case, in which the observation models are completely unknown. This is relevant when the sensor signals are difficult to model or the environment is unknown a priori. Online identification of the observation model requires time and computational resources and might not be feasible, especially on small sensing platforms and in time-critical missions. Due to these complications we also consider a somewhat simpler objective - to control a sensor team in order to localize the source (a single static target) of a physical signal of interest, such as magnetic force, heat, radio signal, or chemical concentration. We assume that the strength of the measured signal is maximal at the location of the source and have the sensors follow its gradient by using a stochastic approximation technique to deal with the underlying noise. We present a single-sensor strategy and a distributed formation-based multi-sensor strategy, which is robust to deformations in the geometry of the sensor team and can be applied to sensors with limited computational resources and no global localization capabilities. Recent model-free source-seeking work which uses a sensor formation to ascend the gradient of a signal field includes Ögren et al. (2004), Wu and Zhang (2011), Li and Guo (2012), and Brinón-Arranz and Schenato (2013). Choi et al. (2009) and Jadaliha et al. (2012) present a general distributed learning and control approach for sensor networks and apply it to source seeking. The sensed signal is modeled by a network of radial basis functions and recursive least squares are used to obtain the model parameters. Ögren et al. (2004) use artificial potentials to decouple the formation stabilization from the gradient ascent. Centralized least-squares are used to estimate the signal gradient. A distributed approach for exploring a scalar field using a cooperative Kalman filter is presented in Zhang and Leonard (2010). The authors design control laws to achieve a formation, which minimizes the estimation error. Similarly, in Brinón-Arranz and Schenato (2013) a circular formation is used to estimate the signal gradient in a distributed manner, based on a Newton-Raphson consensus method. A drawback of these works is the assumption that the sensor formation is maintained perfectly throughout the execution of the algorithm which is hardly possible in a real environment. Our method uses a finite difference scheme to estimate the signal gradient correctly, *even when the sensor formation is not maintained well.* The correct weights necessary to combine the sensor observations into a gradient approximation are recomputed at every measurement location. Instead of a sensor network, a *single* vehicle may travel to several sensing locations in order to collect the same measurements but this requires costly maneuvers to climb the signal gradient effectively (Azuma et al. 2012, Zhang et al. 2007, Liu and Krstić 2010, Stanković and Stipanović 2010, Ghods and Kristić 2011).

This chapter is based on the papers Atanasov et al. (2014d, 2015c, 2012, 2015b) and presents applications in vehicle localization in residential areas, global localization of Google's Project Tango phone (Google ATAP group 2014), active robot localization using object detections, and wireless radio source localization with an unknown sensing model.

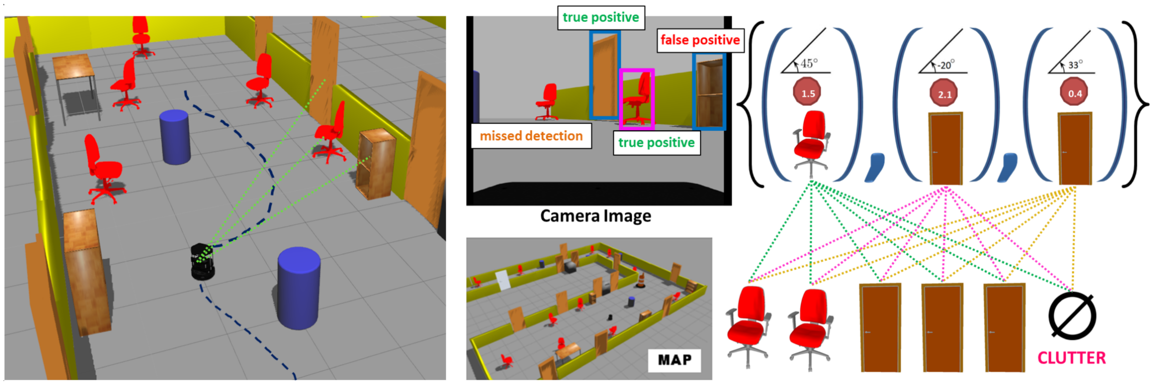


Figure 3.1: A mobile robot (left) localizes itself within a semantic map of the environment by detecting chairs and doors in images (top middle), obtained from its surroundings. A semantic observation received by the robot (top right) consists of a detected class, a detection (confidence) score, and a bearing angle to the detected bounding box. Due to the fact that object recognition misses detections (only one of the two visible chairs is detected) and produces false positives (there is an incorrect door detection), it is appropriate to model the collection of semantic observations via a *set with randomly-varying cardinality.* Finally, correct data association between the object detections (top right) and the landmarks on the prior map (bottom right) plays a key role in the robot's ability to estimate its location.

3.1 Localization from Semantic Observations with Unknown Data Association

In this section, we revisit the active localization problem but this time the robot uses object detections from its surroundings to localize in a prior map of semantically-annotated landmarks. The main challenges are to incorporate the discrete semantic observations in the inference process, to handle the unknown data association between detected and visible objects, and to design a control strategy for active localization, which can handle the nonlinear non-Gaussian inference process. Consider a mobile robot, whose dynamics are governed by the motion model $x_{t+1} = f(x_t, u_t, v_t)$, where $x_t := (x_t^p, x_t^r, x_t^a)$ is the robot state, containing its position x_t^p, orientation x_t^r, and other variables x_t^a, such as velocity and acceleration, u_t is the control input, and v_t is the motion noise. Alternatively, the model can be specified by the probability density function (pdf) of x_{t+1} conditioned on x_t and u_t:

$$p_f(\cdot \mid x_t, u_t). \tag{3.1}$$

The robot has access to a semantic map of the environment, containing n objects with known poses and classes. Let the set $Y = \{y_1, \ldots, y_L\}$ represent the map, where $y_i := (y_i^p, y_i^r, y_i^c)$ consists of the position y_i^p, orientation y_i^r, and class y_i^c of the i-th object. Depending on the application, the object state y_i may capture other observable properties of interest, such as shape priors (Dame et al. 2013).

At each time t, the robot receives data from its sensors and runs an object recognition algorithm, capable of detecting instances from the set of object classes $\mathcal{C}$, present in Y. If some object $y \in Y$ is visible and detected from the current robot pose x_t, then a semantic measurement z_t is obtained. In the remainder, we assume that a semantic measurement, $z_t := (c_t, s_t, b_t)$, consists of a detected class $c_t \in \mathcal{C}$, a detection score $s_t \in \mathcal{S}$, and an estimate

$b_t \in \mathcal{B}$ of the bearing from the sensor to the detected object, where $\mathcal{S}$ is the range of possible scores and $\mathcal{B}$ is the range of bearings, usually specified by the sensor's field of view (e.g., a camera with $\mathcal{B} = [-47^\circ, 47^\circ]$ was used in our experiments). Depending on the sensors and the visual processing, z_t could also contain bounding box, range, color, or other information about the detected object. Detections might also be generated by clutter, which includes the background and any objects not captured on the map Y. Fig. 3.1 illustrates the object recognition process and the challenges associated with it. Due to false alarms and missed detections, a randomly-sized collection of measurements is received at time t. Instead of the traditional vector representation, it is more appropriate to model the collection of semantic observations via a random finite set[3] Z_t. For any t, denote the pdf of the robot state x_t conditioned on the map Y, the past semantic observations $Z_{0:t}$, and the control history $u_{0:t-1}$ by $p_{t|t}$ and that of $x_{t+1} \mid Y, Z_{0:t}, u_{0:t}$ by $p_{t+1|t}$.

Problem (Semantic Localization)**.** *Suppose that control u_t is applied at time $t \geq 0$ and, after moving, the robot obtains a random finite set Z_{t+1} of semantic observations. Given a prior pdf $p_{t|t}$ and the semantic map Y, compute the posterior pdf $p_{t+1|t+1}$ which takes Z_{t+1} and u_t into account.*

Remark. Semantic localization is an inference problem, in which, from the point of view of information acquisition, the robot state is actually the target state to be estimated. However, to be consistent with existing literature we use the notation x_t, instead of the previously-used notation y_t.

It is natural to approach the problem via recursive Bayesian estimation (Appendix C). This, however, requires a sensor observation model, which quantifies the likelihood of a random set Z_{t+1} of semantic observations, conditioned on the set of objects Y and the robot state x_{t+1}. In Sec. 3.1.1, we model the likelihood of an observation received from a single object in the environment. Then, in Sec. 3.1.2, we combine the single-object observation models into an observation model for multiple objects, which captures data association, missed detections, and false alarms. In Sec. 3.1.3, we prove that obtaining the likelihood of a set-valued observation is equivalent to a matrix permanent computation. It is this crucial transformation that enables an efficient polynomial-time solution to the semantic localization problem in Sec 3.1.4 via particle filtering with set-valued observations. Finally, in Sec. 3.1.5, we address the active version of the problem, in which the robot optimizes its trajectory to improve to localization performance.

3.1.1 Semantic Observation Model for a Single Object

The probabilistic model of a semantic observation obtained from a single object consists of three ingredients: a *detection model*, an *observation likelihood*, and a *clutter model*.

The detection model quantifies the probability of detecting an object $y \in Y$ from a given robot state x. Let $\beta(x, y)$ be the true bearing angle from the robot's sensor to the object y in the sensor frame[4]. Let the field of view of the sensor[5] be described by the set

[3] See Mahler (2007, Appendix F) for a formal definition of a random finite set

[4] For example, in 2-D, assuming the robot and the sensor frames coincide, $\beta(x,y) := |\tan^{-1}((x^{p(2)} - y^{p(2)})/(x^{p(1)} - y^{p(1)})) - x^r|$.

[5] The field of view of a camera in 2-D, assuming its frame coincides with the robot's, can be represented

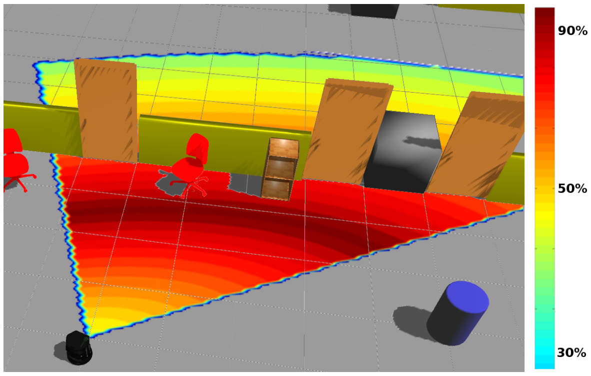


Figure 3.2: Probability of detecting an object within the sensor field of view (not accounting for visibility)

$FoV(x)$. Objects outside the field of view cannot be detected. For the ones within, we use a distance-decaying probability of detection:

$$p_d(y,x) := \begin{cases} p_0 \exp\left\{-\frac{|m_0 - \|y^p - x^p\|_2|}{v_0}\right\} & \text{if } y^p \in FoV(x), \\ 0 & \text{else,} \end{cases} \tag{3.2}$$

where p_0, m_0, v_0 are constants specifying the dependence of the detection probability on distance and are typically learned from training data. The constants might depend on the object's class y^c but this is not explicit to simplify notation. Fig. 3.2 illustrates the detection model. A more complex model which depends on the relative orientation between x and y or uses a different function of their distance is also possible. If visibility information is available from the prior map, it should also be considered when calculating the probability of detection.

When an object $y \in Y$ is detected, the probability of the resulting measurement $z = (c,s,b)$ is quantified by the observation likelihood. Assuming that conditioned on the true object state y, the bearing measurement b is independent of the class c and score s, it is appropriate to model its conditional pdf $p_\beta(\cdot \mid y,x)$ as that of a truncated Gaussian distribution over the bearing range $\mathcal{B}$ with mean $\beta(x,y)$ and covariance Σ_β. The covariance can be learned from training data and can be class dependent. Since object recognition algorithms aim to be scale- and orientation-invariant, we can also assume that the class and score measurements are independent of the robot state x. Then, the observation likelihood of a semantic measurement z can be decomposed as:

$$p_z(z \mid y,x) := p_c(c \mid y^c) p_s(s \mid c, y^c) p_\beta(b \mid y,x), \tag{3.3}$$

where $p_c(c \mid y^c)$ is the confusion matrix of the object detector and $p_s(s \mid c, y^c)$ is a score likelihood. The latter can be learned, for example, by recording the scores from the detected positive examples in a training set and using kernel density estimation (see Fig. 3.9). A more complicated generative model can be used to approximate the observation likelihood p_z. For example, FAB-Map (Cummins and Newman 2008) uses a Chow Liu tree to approximate a probability distribution over visual words learned from SURF descriptors.

by $\{w \in \mathbb{R}^2 \mid \|x^p - w\|_2 \leq r_d, \beta(x,w) \leq \alpha_d\}$, where α_d is the angle of view and r_d is the maximum range at which an object can be detected.

Finally, a model of the pdf, $p_\kappa(z)$, of a false positive measurement generated by clutter is needed. For example, FAB-Map (Cummins and Newman 2008) models the probability that an observation is generated from a place not in the prior map. In our case, $p_\kappa(z)$ is a product of three terms as in (3.3) but it is realistic to assume that the bearing measurement is independent of the robot state and uniformly distributed, i.e., with pdf $1/|\mathcal{B}|$. The class and score likelihoods should be learned from data. Note that the clutter model might also depend on the robot state x and object state y but we suppress this for notational convenience. Our results apply directly to clutter models of the form $p_\kappa(z \mid y, x)$.

3.1.2 Semantic Observation Model for Multiple Objects

In this section, we combine the single-object observation models into a model of the likelihood of a set $Z = \{z_1, \dots, z_m\}$ of semantic observations. Given a robot pose x, let $Y_d(x) := \{y \in Y \mid p_d(y,x) > 0\}$ be the set of objects, detectable from x. In the reminder, we denote the cardinality of Z by m and that of $Y_d(x)$ by n. As mentioned earlier, the data association π between the semantic observations in Z and the visible objects in $Y_d(x)$ is important for constructing the multi-object observation model. The following assumptions are necessary:

(A1) Each measurement $z \in Z$ is generated either by a *single* object y or by clutter.

(A2) An object $y \in Y$ generates either a single detection with probability $p_d(y,x)$ in (3.2) or a missed detection with probability $1 - p_d(y,x)$.

(A3) The process of receiving false-positive measurements is distributed according to the pdf $p_\kappa(z)$ in the measurement space and according to a Poisson distribution with expected value λ over time.

(A4) The false-positive process and the object-detection process are independent and all detections are conditionally independent given the robot and object states.

(A5) Any two measurements in Z are independent conditioned on the robot state x, the detectable objects $Y_d(x)$, and the data association π (a mapping from the visible objects $Y_d(x)$ to the measurement set Z).

We specify the pdf of Z, conditioned on x and $Y_d(x)$, in five steps of increasing complexity.

All measurements are false positive

The simplest case is when there are no objects in proximity to the sensor, i.e., $Y_d(x) = \emptyset$. Then, any generated measurements would be from clutter. The correct observation model in this case is due to assumption (A3) of a Poisson false-positive process:

$$p(Z \mid \emptyset, x) = \frac{e^{-\lambda}\lambda^{|Z|}}{|Z|!} \prod_{z \in Z} p_\kappa(z). \tag{3.4}$$

This integrates to 1 if we use the set integral definition in Mahler (2007, Ch.11.3.3):

$$\int p(Z)dZ := \sum_{m=0}^{\infty} \int p(\{z_1, \dots, z_m\})dz_1 \dots dz_m.$$

No missed detections and no false positives

Next, suppose that there are detectable objects in proximity to the sensor but let the detection be perfect. In other words, assume that every detectable object generates a measurement, i.e., $p_d(y, x) = 1$ for any $y \in FoV(x)$, and no measurements arise in any other way, i.e., $\lambda = 0$. If the number of measurements m is not equal to the cardinality n of the set of detectable objects $Y_d(x)$, then $p(Z \mid Y_d(x), x) = 0$. Otherwise, the main challenge is identifying the correct data association π. In other words, it is not clear which of the detectable objects $Y_d(x)$ on the map produced which of the measurements in Z.

More formally, let $\Pi_{n,m}$ be the set of *one-to-one* functions $g : \{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$ with $n \leq m$. Due to the "perfect detection" assumption, $m = n$ and a particular data association can be represented by a mapping $\pi \in \Pi_{n,n}$ from the set of detectable objects to the set of measurements. The data association π is just a permutation of $\{1, \ldots, n\}$ but it is not clear which of the possible $|\Pi_{n,n}| = n!$ associations is the correct one. If a particular data association π is chosen, it is straightforward to combine the single-object observation likelihoods in (3.3) via the independence assumptions (A4), (A5) to obtain the pdf of Z:

$$p(Z \mid Y_d(x), x, \pi) = \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_i, x),$$

where $\{y_1, \ldots, y_n\}$ is an enumeration of $Y_d(x)$. Assuming a uniform prior[6] on the possible data associations:

$$p(\pi \mid Y_d(x), x) = \frac{1}{n!}, \quad \pi \in \Pi_{n,n},$$

existing work (e.g., FastSLAM by Montemerlo and Thrun (2003)) resorts to maximum-likelihood estimation and computes the likelihood of Z as follows:

$$p(Z \mid Y_d(x), x) = \frac{1}{n!} \max_{\pi \in \Pi_{n,n}} \left(\prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_i, x) \right).$$

The above equality, however, disagrees with the law of total probability, which states that the data association should be marginalized. The observation model in the "perfect detection" case is:

$$p(Z \mid Y_d(x), x) = \sum_{\pi \in \Pi_{n,n}} p(Z \mid Y_d(x), x, \pi) p(\pi \mid Y_d(x), x) = \frac{1}{n!} \sum_{\pi \in \Pi_{n,n}} \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_i, x). \quad (3.5)$$

Intuitively, (3.5) is quantifying the likelihood of Z by averaging the likelihoods of the individual measurements over all possible data associations. The reason, why existing work avoids this marginalization, is that the summation over all $n!$ data associations is computationally demanding. However, in Sec. 3.1.3, we will present an efficient method for computing (3.5). Before that, we relax the perfect-detection assumption.

[6] It is possible to track the data association distribution over time (Bar-Shalom et al. 2009).

No false positives but missed detections are possible

Now, suppose that some of the objects in proximity to the sensor might not be detected. Assuming no false positives still, the number of measurements m should be at most the number of detectable objects n, i.e., if $m > n$, then $p(Z \mid Y_d(x), x) = 0$. In the case that $m \le n$, we have $\pi \in \Pi_{m,n}$ and there are $|\Pi_{m,n}| = {}_nP_m := \frac{n!}{(n-m)!}$ possible data associations. Let $\mathcal{D}(\pi) := \cup_{j=1}^m \{\pi(j)\}$ be the set of true-positive detections according to π and $\mathcal{M}(\pi) := \{1, \dots, n\} \setminus \mathcal{D}(\pi)$ be the set of missed detections. Finally, let $\mathcal{A}(\pi)$ be the event that the true-positive detections $\mathcal{D}(\pi)$ are assigned to the measurements in Z in the way specified by π. Then, we can quantify the likelihood of $\pi \in \Pi_{m,n}$, using the detection model (3.2), as follows:

$$\begin{aligned}
p(\pi \mid Y_d(x), x) &= \mathbb{P}(\mathcal{A}(\pi)) \times \mathbb{P}(\{y_i \mid i \in \mathcal{D}(\pi)\} \text{ are detected}) \times \mathbb{P}(\{y_i \mid i \in \mathcal{M}(\pi)\} \text{ are missed}) \\
&= \frac{1}{m!} \prod_{j=1}^{m} p_d(y_{\pi(j)}, x) \prod_{i \in \mathcal{M}(\pi)} (1 - p_d(y_i, x)).
\end{aligned}$$

See Appendix D.11 for a verification that $p(\pi \mid Y_d(x), x)$ is a valid pdf. As before, we can derive the likelihood of Z by marginalizing the data association:

$$\begin{aligned}
p(Z \mid Y_d(x), x) &= \sum_{\pi \in \Pi_{m,n}} p(Z \mid Y_d(x), x, \pi) p(\pi \mid Y_d(x), x) \\
&= \sum_{\pi \in \Pi_{m,n}} \left[\prod_{j=1}^{m} p_z(z_j \mid y_{\pi(j)}, x) \right] \left[\frac{1}{m!} \prod_{j=1}^{m} p_d(y_{\pi(j)}, x) \prod_{i \in \mathcal{M}(\pi)} (1 - p_d(y_i, x)) \right] \\
&= \frac{1}{m!} \prod_{i=1}^{n} (1 - p_d(y_i, x)) \sum_{\pi \in \Pi_{m,n}} \prod_{j=1}^{m} \frac{p_d(y_{\pi(j)}, x) p_z(z_j \mid y_{\pi(j)}, x)}{1 - p_d(y_{\pi(j)}, x)} \qquad (3.6)
\end{aligned}$$

The observation model is similar to the "perfect detection" case in (3.5) but the single-object-measurement likelihoods need to be scaled by the probabilities of detection. If no measurements are received but $Y_d(x) \neq \emptyset$, the above simplifies to:

$$p(\emptyset \mid Y_d(x), x) = \prod_{i=1}^{n} (1 - p_d(y_i, x)) \qquad (3.7)$$

No missed detections but false positives are possible

In this case, $n \le m$ (otherwise $p(Z \mid Y_d(x), x) = 0$) and $\pi \in \Pi_{n,m}$. Again, let $\mathcal{A}(\pi)$ be the event that the true-positive detections $(Y_d(x))$ are assigned to the measurements in Z in the particular way specified by π. The likelihood of π is:

$$\begin{aligned}
p(\pi \mid Y_d(x), x) &= \mathbb{P}(\mathcal{A}(\pi)) \times \mathbb{P}(\{n \text{ true positives}\}) \times \mathbb{P}(\{m - n \text{ false positives}\}) \\
&= \frac{1}{{}_mP_n} \times 1 \times \frac{e^{-\lambda} \lambda^{m-n}}{(m-n)!},
\end{aligned}$$

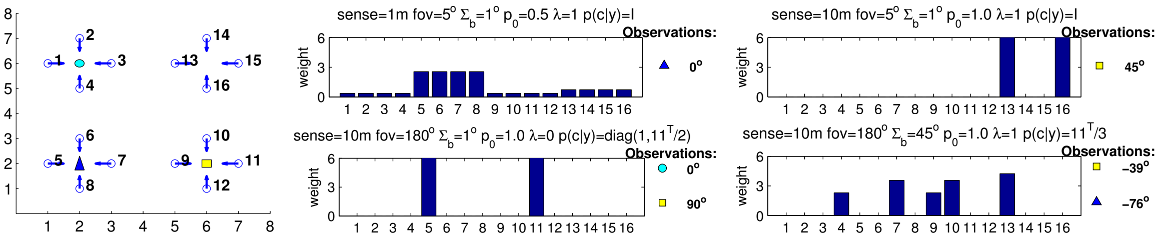


Figure 3.3: Consider a localization scenario with 16 possible poses, indicated by the arrows on the left-most plot. There are three objects in the environment: a yellow square (class 1), a cyan circle (class 2), and a blue triangle (class 3). Initially, the 16 poses are equally likely (each has weight 1). Suppose that only one set of semantic observations is received. The four plots to the right show how the likelihoods of the 16 locations change, depending on the received set. At each location, the likelihood of the semantic observation set is computed via (3.9) and normalized, so that the sum of the likelihoods is 16. The parameters, used in the semantic observation model, are listed at the top of the plots. For simplicity, the semantic observations here contain only bearing and class information. In the top right plot since the field of view is only 5° it is not possible to observe a yellow square at a 45° bearing from poses $9-12$. Also, since the sensing range is 10 m and there are no missed detections, poses $1-8$ and $14-15$ are not possible.

which is a valid pdf (see Appendix D.11). The likelihood of Z is obtained by marginalizing the data association:

$$\begin{aligned}
p(Z \mid Y_d(x), x) &= \sum_{\pi \in \Pi_{n,m}} p(Z \mid Y_d(x), x, \pi) p(\pi \mid Y_d(x), x) \\
&= \sum_{\pi \in \Pi_{n,m}} \prod_{i=1}^{n} p_z(z_{\pi(i)} \mid y_i, x) \prod_{j \in \{1,\dots,m\} \setminus \cup_{i=1}^{n} \{\pi(i)\}} p_k(z_j) \frac{e^{-\lambda} \lambda^{m-n}}{m!} \\
&= \frac{e^{-\lambda} \lambda^{m}}{m!} \prod_{j=1}^{m} p_k(z_j) \sum_{\pi \in \Pi_{n,m}} \prod_{i=1}^{n} \frac{p_z(z_{\pi(i)} \mid y_i, x)}{\lambda p_k(z_{\pi(i)})}
\end{aligned} \tag{3.8}$$

Both missed detections and false positives are possible

Finally, consider the most general case that captures all nuisances of object recognition: missed detections, false positives, and unknown data association. If there are no detectable objects close by ($n = 0$), then the pdf of Z is given by (3.4). If no measurements are received ($m = 0$), then the pdf of Z is given by (3.7). Otherwise, $\pi \in \bar{\Pi}_{n,m}$, where $\bar{\Pi}_{n,m}$ is the set of functions $g : \{1, \dots, n\} \to \{0, 1, \dots, m\}$ with the property: $g(i) = g(i') > 0 \Rightarrow i = i'$, which ensures that (A1) is satisfied. The index '0' in the range of g represents the case of missing a detectable object. For example, it allows for the possibility that all detectable objects are missed (associated with '0'), in which case we obtain the term in (3.7). The number of possible data associations now is

$$|\bar{\Pi}_{n,m}| = \sum_{k=0}^{\min\{n,m\}} \binom{n}{k} \, {}_mP_k,$$

where the index k indicates the number of true-positive assignments made by a particular data association π. The likelihood of $\pi \in \bar{\Pi}_{n,m}$ with k true-positive assignments is:

$$\begin{aligned} p(\pi \mid Y_d(x), x) = \mathbb{P}(\mathcal{A}(\pi)) \times \mathbb{P}(\{y_i \mid \pi(i) > 0\} \text{ are detected}) \times \\ \mathbb{P}(\{y_i \mid \pi(i) = 0\} \text{ are missed}) \times \mathbb{P}(\{m-k \text{ false positives}\}) \\ = \frac{1}{{}_mP_k} \prod_{i:\pi(i)>0} p_d(y_i, x) \prod_{i:\pi(i)=0} (1 - p_d(y_i, x)) \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!}, \end{aligned}$$

where, as before, $\mathcal{A}(\pi)$ is the event that the k true-positive detections are assigned to the measurements in Z in the particular way specified by π. See Appendix D.11 for a verification that $p(\pi \mid Y_d(x), x)$ is a valid pdf. As before, we can derive the likelihood of Z by marginalizing the data association:

$$\begin{aligned} p(Z \mid Y_d(x), x) &= \sum_{\pi \in \bar{\Pi}_{n,m}} p(Z \mid Y_d(x), x, \pi) p(\pi \mid Y_d(x), x) \\ &= \sum_{\pi \in \bar{\Pi}_{n,m}} \left[\prod_{i:\pi(i)>0} p_z(z_{\pi(i)} \mid y_i, x) \prod_{j \in \{1,\ldots,m\} \setminus \cup_{i=1}^{n} \{\pi(i)\}} p_\kappa(z_j) \right] p(\pi \mid Y_d(x), x) \\ &= p(Z \mid \emptyset, x) p(\emptyset \mid Y_d(x), x) \sum_{\pi \in \bar{\Pi}_{n,m}} \prod_{i:\pi(i)>0} \frac{p_d(y_i, x) p_z(z_{\pi(i)} \mid y_i, x)}{(1 - p_d(y_i, x)) \lambda p_\kappa(z_{\pi(i)})}. \end{aligned} \tag{3.9}$$

To gain intuition about the observation model in this most general case, refer to Fig. 3.3.

3.1.3 Connection with the Matrix Permanent

As mentioned earlier, it is not apparent how to efficiently compute the sum over all data associations π required in multi-object semantic observation model (3.9). To gain intuition we begin with the simpler case of "perfect detection" in (3.5). Fix a robot state x and consider the nontrivial case when the received measurements Z and the detectable landmarks $Y_d(x)$ have the same cardinality m. We can think about the data association between Z and $Y_d(x)$ from a graph-theoretic perspective. Represent the sets $V_1 := Y_d(x)$ and $V_2 := Z$ by the vertices of a complete balanced bipartite graph and let E be the edge set. Associate the weight $w_e := p_z(z \mid y, x)$ with every edge $e := (z, y) \in E$ and consider the weighted bipartite graph $G := (V_1, V_2, E, w)$. The data associations π, between the objects V_1 and the measurements V_2, in the "perfect detection" case (3.5), in fact, correspond to *perfect matchings*[7] in G. Given a perfect matching π, the product term inside the sum in (3.5) corresponds to its weight. Then, the sum over all π corresponds to the sum of the weights of all perfect matchings in G, which notably is equal to the permanent of the adjacency matrix of G.

[7] A matching in graph G is a subgraph of G in which no two edges share a common vertex. The weight of a matching is the product of all its edge weights. A matching is perfect if it contains all of G's vertices.

Permanent. *The permanent of an $n \times m$ matrix $A = [A(i,j)]$ with $n \leq m$ is defined as:*

$$\mathbf{per}(A) := \sum_{\pi} \prod_{i=1}^{n} A(i, \pi(i)),$$

where the sum is over all one-to-one functions $\pi : \{1, \dots, n\} \to \{1, \dots, m\}$. If $n > m$, then $\mathbf{per}(A) := \mathbf{per}(A^T)$.

It is now clear that the likelihood of a set of semantic observations in the "perfect detection" case can be obtained by computing the permanent of a matrix.

Theorem 3.1. *The likelihood in (3.5) of a random finite set of semantic observations, $Z = \{z_1, \dots, z_m\}$, in the case of no false positives and no missed detections, with $n := |Y_d(x)| = m$ detectable objects, satisfies:*

$$p(Z \mid Y_d(x), x) = \frac{1}{n!} \mathbf{per}(Q),$$

where Q is a $n \times n$ matrix with $Q(i,j) := p_z(z_j \mid y_i, x)$ and $\{y_1, \dots, y_n\}$ is an enumeration of the set $Y_d(x)$ of detectable objects.

The general case in (3.9), where both false positives and missed detections are possible, can be analyzed using the same graph-matching intuition.

Theorem 3.2. *Given a robot state x and set of detectable objects $Y_d(x)$ with $n := |Y_d(x)| > 0$, the likelihood of a random finite set $Z = \{z_1, \dots, z_m\}$ of semantic observations, with $m > 0$, when both false-positive and missed detections are possible satisfies:*

$$\begin{aligned} p(Z \mid Y_d(x), x) = {} & \frac{e^{-\lambda}\lambda^m}{m!} \prod_{z \in Z} p_\kappa(z) \prod_{y \in Y_d(x)} (1 - p_d(y, x)) \\ & \times \begin{cases} \frac{1}{m!}\,\mathbf{per}\left(\begin{bmatrix} Q & I_n \\ 1_{m,m} & 1_{m,n} \end{bmatrix}\right), & n \leq m \\ \\ \frac{1}{n!}\,\mathbf{per}\left(\begin{bmatrix} Q^T & I_m \\ 1_{n,n} & 1_{n,m} \end{bmatrix}\right), & m \leq n \end{cases} \end{aligned} \tag{3.10}$$

where $p_d(y, x)$ is the probability of detecting object $y \in Y_d(x)$, λ is the expected number of false-positive detections with spatial pdf $p_\kappa(\cdot)$, $1_{n,m}$ is a $n \times m$ matrix of all ones, and Q is a matrix with elements:

$$Q(i,j) := \frac{p_d(y_i, x) p_z(z_j \mid y_i, x)}{(1 - p_d(y_i, x))\lambda p_\kappa(z_j)}, \qquad \begin{aligned} i &= 1, \dots, n, \\ j &= 1, \dots, m. \end{aligned}$$

Connections between the matrix permanent and data association have been identified in the target tracking community (Oh et al. (2009), Collins and Uhlmann (1992), Pasula et al. (1999), Morelande (2009), Liggins et al. (2008, Ch.11)) but this is the first connection with the random-finite-set observation model. Thm. 3.2 maps the problem of determining the pdf of Z in the general case in (3.9) to the problem of finding the permanent of a

$(m+n) \times (m+n)$ square matrix. The problem is still computationally challenging because computing the permanent of a matrix is #P-complete[8] (Valiant 1979). However, Thm. 3.2 allows us to leverage the extensive literature on approximation algorithms for computing the matrix permanent. The proof of Thm. 3.2 (see Appendix D.12) reduces the problem of summing the weights of all matchings in an unbalanced bipartite graph to the problem of summing the weights of all perfect matchings in an unbalanced bipartite graph and then to the problem of summing the weights of all perfect matchings in a balanced bipartite graph. We could stop at the first reduction, which would require calculating the permanent of a rectangular matrix. The reason for the second reduction is that existing permanent-approximation algorithms are much better for square than for rectangular matrices.

An exact method for computing the permanent of a $d \times d$ matrix, proposed by Ryser (1963) and later improved by Nijenhuis and Wilf (1978, Ch.23), is summarized in Alg. 6. Its time complexity is $\Theta(d2^{d-1})$. The dimension of the matrix in (3.10) is equal to the number of detections returned by the vision algorithm plus the number of detectable objects within the sensor field of view, which in some cases is often small enough to enable a real-time implementation of Alg. 6. Otherwise, there are a number of polynomial-time arbitrarily-close approximations to the permanent computation. For example, Jerrum et al. (2004) show that for any $\epsilon \in (0,1]$ and $\delta > 0$, there exists a randomized algorithm whose output comes within a factor $(1 \pm \epsilon)$ of $\mathbf{per}(A)$ with probability at least $1-\delta$ with a random running time T such that $\mathbb{E}(T) = O(d^{10}(\log d)^3)$. The running time was later improved by Bezáková et al. (2006) to $O(d^7(\log d)^4)$. Also, when $A \in [0,1]^{d \times d}$ is a matrix such that all row and column sums are at least γd for $\gamma \in (0.6, 1]$, Law (2009, Ch.2.2) provides an algorithm with expected running time $O(d^4(\log d + \epsilon^{-2} \log \delta^{-1}))$.

Proposition 3.3. *Given m object detections and n visible objects, the complexity of computing the likelihood in* (3.9) *of the semantic observation set is $O((m+n)2^{(m+n)})$, if computed exactly via Alg. 6, and $O((m+n)^7(\log(m+n))^4)$ in expectation, if computed approximately via the randomized method of Bezáková et al. (2006).*

Using the idea of Thm 3.2, similar results can be obtained for the simpler cases with no false-positives or no missed detections. Appendix D.13 shows the link between the likelihood of a set of semantic observations and the matrix permanent for all cases.

3.1.4 Semantic Localization

Now that we have a sensor observation model for multiple objects (3.9) and an efficient way to compute it (Prop. 3.3), we can return to the semantic localization problem.

Proposition 3.4. *The Bayesian recursion that solves the semantic localization problem is:*

$$\begin{aligned} &\textbf{\textit{Predict:}} \quad p_{t+1|t}(x) = \int p_f(x \mid x', u_t) p_{t|t}(x') dx' \\ &\textbf{\textit{Update:}} \quad p_{t+1|t+1}(x) = \eta_{t+1} p(Z_{t+1} \mid Y_d(x), x) p_{t+1|t}(x), \end{aligned} \tag{3.11}$$

where $p(Z_{t+1} \mid Y_d(x), x)$ is the random finite set observation model in (3.9) and η_{t+1} is a normalization constant.

[8] A #P-complete problem is equivalent to computing the number of accepting paths of a polynomial-time nondeterministic Turing machine and #P contains NP.

Algorithm 6 Permanent (Nijenhuis and Wilf 1978, Ch.23)

```
Input: $d \times d$ matrix $A$     Output: $per(A)$
for $i = 1, \dots, d$ do
    $x(i) \leftarrow A(i,d) - \frac{1}{2}\sum_{j=1}^{d} A(i,j)$
$s \leftarrow -1$,     $g \leftarrow$ false$(d,1)$,     $p \leftarrow s\prod_{i=1}^{d} x(i)$
for $k = 2, \dots, 2^{d-1}$ do
    if $k$ is even then $j \leftarrow 1$                                  ▷ Obtain next gray code subset
    else {   $j \leftarrow 2$
             while $g_{j-1}$ is false do
                   $j \leftarrow j+1$          }
    $s \leftarrow -s$,     $z \leftarrow 1 - 2g_j$,     $g_j \leftarrow$ not $g_j$
    for $i = 1, \dots, d$ do
        $x(i) \leftarrow x(i) + zA(i,j)$
    $p \leftarrow p + s\prod_{i=1}^{d} x(i)$
return $2(-1)^d p$
```

Like its vector-based counterpart, an exact implementation of the Bayes filter with set-valued observations is intractable. The particle filter (Appendix C.3) is an approximation to the Bayes filter with vector-valued observations, which has been very successful in practice for geometric localization (Thrun et al. 2005, Ch.4). It can be applied in this case too, with the exception that, instead of the conventional vector-based measurement update, the particle weights need to be updated with the likelihood of the received semantic observation sets Z_t. Since the robot state is still vector-valued, we represent its pdf $p_{t|t}$ at time t with a set of particles $\{w^k_{t|t}, x^k_{t|t}\}_{k=1}^N$:

$$p_{t|t}(x) \approx \sum_{k=1}^{N} w^k_{t|t}\delta(x - x^k_{t|t}),$$

where $\delta(\cdot)$ is a Dirac delta function. The particle-filter implementation of (3.11), with the motion model (3.1) as a proposal distribution and the semantic observation model (3.9) as a measurement update, is summarized in Alg. 7. A matrix permanent algorithm, such as Alg. 6, is needed to update the particle weights in line 6.

Algorithm 7 Set-based Particle Filter

```
1: Input: Particle set $\{w^k_{t|t}, x^k_{t|t}\}_{k=1}^N$, motion model pdf $p_f$, observation model pdf $p$, semantic map $Y$,
   control input $u_t$, detection set $Z_{t+1}$
2: Output: Particle set $\{w^k_{t+1|t+1}, x^k_{t+1|t+1}\}_{k=1}^N$
3: for $k = 1, \dots, N$ do
4:     Predict: Draw $x^k_{t+1|t}$ from pdf $p_f(\cdot \mid x^k_{t|t}, u_t)$
5:              $w^k_{t+1|t} \leftarrow w^k_{t|t}$
6:     Update: $w^k_{t+1|t+1} \leftarrow p\big(Z_{t+1} \mid Y_d(x^k_{t+1|t}), x^k_{t+1|t}\big)w^k_{t+1|t}$
7:              $x^k_{t+1|t+1} \leftarrow x^k_{t+1|t}$
8: Normalize the weights $\{w^k_{t+1|t+1}\}_{k=1}^N$ and re-sample if necessary
```

3.1.5 Active Semantic Localization

In this section, we return to the active information acquisition theme and emphasize that the robot can plan its motion in order to improve the performance of the semantic localization. As before, the main idea is to choose a sequence $\sigma := u_t, \ldots, u_{t+T-1}$ of control inputs for the next T time steps in order to minimize some measure of uncertainty in the pose. To simplify the notation going forward, assume, without loss of generality, that the current time is $t = 0$. As before, we use the entropy $\mathbf{H}(x_{0:T} \mid Z_{1:T})$ of the current and future poses $x_{0:T}$ conditioned on the future semantic observations $Z_{1:T}$ as the uncertainty criterion. Note that the problem is different from the discussion in Ch. 2, since now the target inference is done via a particle filter instead of a Kalman filter. The main complication is that the entropy criterion is no longer independent of the measurement realization and the separation principle (Thm. 2.1) does not hold anymore. Hence, closed-loop planning is needed to solve the resulting stochastic active information acquisition problem. In this chapter, we resort to greedy planning to avoid the difficult nonmyopic closed-loop planning problem. We will return to nonmyopic closed-loop information acquisition in Ch. 4.

Instead of computing a control policy that depends on the measurement realizations, we choose a set Σ of admissible control sequences, evaluate the entropy criterion for each sequence, and pick the control sequence that minimizes entropy. As we show below, evaluating the entropy criterion for the set-based particle filter even a single time is computationally very challenging. This is in contrast with Ch. 2, where, due to the Kalman filter inference process, the entropy criterion was proportional to the log-determinant of target covariance matrix and was much easier to evaluate.

Problem (Active Semantic Localization)**.** *Given a prior pdf of the pose x_0, the semantic map Y, and a space $\Sigma := \{\sigma^i \mid \sigma^i := u_0^i, \ldots, u_{T-1}^i\}_{i=1}^M$ of possible control sequences of length T, choose the sequence σ^*, which minimizes the entropy of the current and future poses $x_{0:T}$, conditioned on the future semantic observations $Z_{1:T}$:*

$$\begin{aligned} \sigma^* \in \arg\min_{\sigma \in \Sigma} \quad & \mathbf{H}(x_{0:T} \mid Z_{1:T}) \\ s.t. \quad & x_{s+1} \sim p_f(\cdot \mid x_s, \sigma_s), \;\; s = 0, \ldots, T-1 \\ & Z_s \sim p(\cdot \mid Y_d(x_s), x_s), \;\; s = 1, \ldots, T \end{aligned} \tag{3.12}$$

where p_f is the motion model (3.1) *and $p(\cdot \mid Y_d(x), x)$ is the semantic observation model* (3.10)*.*

We assume that Σ has been designed offline and consists of motion primitives, each with T sampling poses, that can provide reasonable coverage of the robot's surroundings. Sixty locations with outward facing orientations were chosen on the perimeter of a circle of radius 10 m. A differential-drive controller was used to to generate a control sequence of length $T = 5$, which would lead a robot at the origin to each of the selected locations. Fig. 3.4 shows the resulting set of motion primitives.

While accuracy is very important for the inference process, speed is crucial during planning. At each time step, the inference process needs to be carried out for a single set of observations (the actual one) but the planning process needs to envision various measurement realizations. Real-time planning requires very efficient computation of the

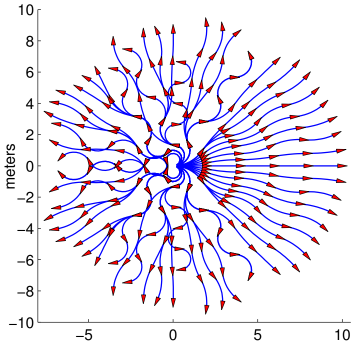


Figure 3.4: The set of motion primitives used for active localization. Each segment contains 5 measurement poses indicated by the red triangles.

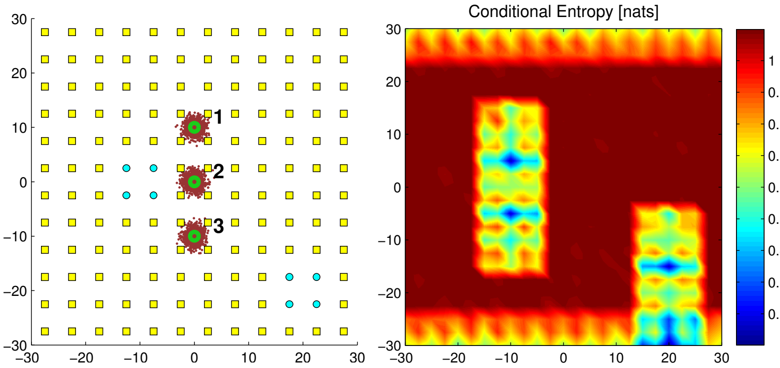


Figure 3.5: The left plot shows a simulation of a 2-D localization scenario with two object classes (circle, square). The prior density of the observer's pose is represented by the dark red particle set, which is concentrated in 3 locations (green). The observer has a field of view of 360° and a sensing range of $4m$. The other parameters of the observation model were $p_0 = 0.73, m_0 = 2.7, v_0 = 35, \Sigma_\beta = 4^\circ, \lambda = 0.5$. The right plot shows the entropy of the observer's location (in the local frame of reference) conditioned on one set of semantic observations. As the summarized particle set contains only 3 particles, the entropy varies from 0 to 1.099 nats.

objective function in (3.12). Given a control sequence $u_{0:T-1}$, the conditional entropy is defined as follows:

$$\mathbf{H}(x_{0:T} \,|\, Z_{1:T}) = \int \left[\int -g(x_{0:T}, Z_{1:T}, u_{0:T-1}) \log g(x_{0:T}, Z_{1:T}, u_{0:T-1}) dx_{0:T} \right] p(Z_{1:T}) dZ_{1:T} \quad (3.13)$$

where $p(Z_{1:T})$ is the (not conditional) pdf of the semantic observations and we have defined:

$$g(x_{0:T}, Z_{1:T}, u_{0:T-1}) := \frac{p_{0|0}(x_0) \prod_{s=1}^{T} p_f(x_s \mid x_{s-1}, u_{s-1}) \prod_{s=1}^{T} p(Z_s \mid Y_d(x_s), x_s)}{p(Z_{1:T})}, \qquad (3.14)$$

using the assumption that the sets $Z_{1:T}$ are conditionally independent, given the set of detectable objects $\bigcup_{s=1}^{T} Y_d(x_s)$ and the trajectory $x_{1:T}$. Even if the measurement sets $Z_{1:T}$ were given, the inside integral in (3.13) would need to be evaluated for each of the M control sequences with N^{T+1} future particle evolutions, each requiring T evaluations (permanent computations) of the semantic observation likelihood. Assuming exact permanent computations, this makes the complexity of obtaining just the inside integral: $O(MN^{T+1} \sum_{s=1}^{T} (|Y| + |Z_s|) 2^{|Y|+|Z_s|})$! In order to address the complexity of this planning problem we will use several approximations.

Maximum likelihood data association

First, we resort to maximum likelihood data association during planning to avoid expensive permanent computations. Given a set Z of semantic observations with $m := |Z|$, for each particle x, we construct an association function $\pi : \{1, \ldots, m\} \to \{0, 1, \ldots, |Y_d(x)|\}$ by processing the measurements $z_j,\ j = 1, \ldots, m$ sequentially. For z_j, we compute:

$$\max\left\{\max_{i\in\{1,\ldots,|Y_d(x)|\}} p_d(y_i, x) p_z(z_j \mid y_i, x), \frac{\lambda}{|Z| - q} p_\kappa(z_j)\right\},$$

where q is the number of measurements already assigned to clutter, and let $\pi(j) = i$, if the max is achieved at a detectable object $y_i \in Y_d(x)$, and $\pi(j) = 0$, otherwise. Then, the likelihood of Z is:

$$p(Z \mid Y_d(x), x, \pi) = \frac{e^{-\lambda}}{|Z|!} \prod_{j|\pi(j)=0} \lambda p_\kappa(z_j) \prod_{y\in D} (1 - p_d(y, x)) \prod_{j|\pi(j)>0} p_d(y_{\pi(j)}, x) p_z(z_j \mid y_{\pi(j)}, x),$$

where D is the set of unassigned detectable objects. The use of maximum likelihood data association in (3.14) replaces the $O(\sum_{s=1}^{T}(|Y| + |Z_s|)2^{|Y|+|Z_s|})$ cost of permanent computations by $O(\sum_{s=1}^{T} |Z_s||Y|)$.

Noiseless motion

Another problematic term in the complexity characterization of the inner integral in (3.13) is N^{T+1}. It is due to the evolution of the set of N particles over the planning horizon T. The integral can be simplified significantly by neglecting the noise in the motion model (3.1). In other words, the robot can be optimistic and plan its future trajectories with a "perfect motion" assumption (albeit not satisfied in reality):

$$p_f(x_{s+1} \mid x_s, u_s) = \delta(x_{s+1} - f(x_s, u_s, 0)). \tag{3.15}$$

For the given control sequence $u_{0:T-1}$, let the (now) deterministic evolution of the particles in the initial particle set $\{w_{0|0}^k, x_{0|0}^k\}$ over the time horizon $s = 0, \ldots, T-1$ be $x_{s+1|s+1}^k := f(x_{s|s}^k, u_s, 0)$. Then, the "perfect motion" assumption implies that

$$g(x_{0:T}, Z_{1:T}, u_{0:T-1}) = \sum_{k=1}^{N} \frac{w_{0|0}^k \prod_{s=1}^{T} p(Z_s \mid Y_d(x_{s|s}^k), x_{s|s}^k)}{p(Z_{1:T})} \prod_{s=0}^{T} \delta(x_{s|s}^k - x_s),$$

which in turn reduces the integral in (3.13) to:

$$H(x_{0:T} \mid Z_{1:T}) = \int \left[-\sum_{k=1}^{N} \tilde{w}^k(Z_{1:T}) \log \tilde{w}^k(Z_{1:T})\right] p(Z_{1:T}) dZ_{1:T}, \tag{3.16}$$

where

$$\tilde{w}^k(Z_{1:T}) := \frac{w_{0|0}^k \prod_{s=1}^{T} p(Z_s \mid Y_d(x_{s|s}^k), x_{s|s}^k)}{p(Z_{1:T})} \tag{3.17}$$

are the normalized weights of the (updated) particle set at time T. Note that $p(Z_{1:T})$ is a normalization factor and does not need to be computed explicitly. Combining the result in (3.16) with maximum likelihood data association, reduces the computational complexity of the inner integral (now a sum) in (3.13) from $O(MN^{T+1} \sum_{s=1}^{T}(|Y| + |Z_s|)2^{|Y|+|Z_s|})$ to $O(MN|Y| \sum_{s=1}^{T} |Z_s|)$. Most importantly, the new complexity does not have an exponential dependence on the problem parameters.

Particle set summarization

A final reduction in complexity can be obtained by decreasing the number of particles that represent the prior pose distribution. For planning purposes, it is not crucial to represent the distribution accurately but rather it is important that it contains the competing hypotheses. Charrow et al. (2013) proposed replacing subsets of similar particles with their average, in the context of target tracking with range-only sensing. The authors prove that, for Gaussian measurement noise, the approximation introduces a bounded error in the mutual information between the observations and the target state. We adopt the same idea here, despite that the measurement noise (except for the bearing noise) is not Gaussian. Specifically, we partition the robot state space with a regular square grid and replace particles, contained in the same cell, with their weighted average. Depending on the size of grid cells, this approximation can reduce the number of particles significantly (see Fig. 3.5). We emphasize that all these approximations (particle summarization, noiseless motion, maximum likelihood data association) are used only in the planning process. The inference process still uses the full particle set, the full semantic observation model in (3.10), and considers motion noise.

Now that the evaluation of the inner integral in (3.13) has been simplified significantly, we consider the outside integration over the set-valued variables $Z_{1:T}$. Since not even the cardinality of the measurement sets is known, an exact computation would be hopeless. However, given a robot trajectory and the semantic map Y, the semantic observation model (3.10) can be used to simulate measurements from the detectable objects and, in turn, obtain a Monte Carlo approximation to (3.13).

Monte Carlo integration

The key to a fast and accurate Monte Carlo approximation of (3.13) is to simulate measurement sets from $p(Z_{1:T})$ in a way that the samples are concentrated in regions that make large contributions to the integral. Observe that, due to the particle set approximation of the prior $p_{0|0}$ and the "perfect motion" assumption, $p(Z_{1:T})$ is a finite mixture model:

$$\begin{aligned} p(Z_{1:T}) &= \int p(Z_{1:T}, x_{0:T}) dx_{0:T} = \int \prod_{s=1}^{T} p(Z_s \mid Y_d(x^k_{s|s}), x^k_{s|s}) \delta(x^k_{s|s} - x_s) p_{0|0}(x_0) dx_{0:T} \\ &= \sum_{k=1}^{N} w^k_{0|0} \prod_{s=1}^{T} p(Z_s \mid Y_d(x^k_{s|s}), x^k_{s|s}). \end{aligned}$$

An efficient way to sample from the mixture model $p(Z_{1:T})$ is to first sample the mixture component, according to the weight w^k, and then sample each Z_s from the conditional densities. This has the additional benefit of sampling observation sets that are likely to be encountered by the robot and should provide a large contribution to the integral. Thus, for a given control sequence $u_{0:T-1}$, we follow the following steps to estimate $\mathbf{H}(x_{0:T} \mid Z_{1:T})$:

1. Sample a particle $x^l_{0|0}$ from the prior particle set according to the weights $w^k_{0|0}$, $k = 1, \ldots, N$

2. Compute the particle trajectory $x^l_{s+1|s+1} := f(x^l_{s|s}, u_s, 0)$ for $s = 0, \ldots, T-1$

3. Sample Z_s^l from the semantic observation model $p(Z_s^l \mid Y_d(x_{s|s}^l), x_{s|s}^l)$ for $s = 1, \ldots, T$
4. Compute the normalized updated particle weights $\tilde{w}^k(Z_{1:T}^l)$ via (3.17) for $k = 1, \ldots, N$
5. Evaluate the inner sum: $H_l := -\sum_{k=1}^N \tilde{w}^k(Z_{1:T}^l) \log \tilde{w}^k(Z_{1:T}^l)$
6. Repeat the above steps N_z times to obtain the Monte Carlo approximation:

$$\mathbf{H}(x_{0:T} \mid Z_{1:T}) \approx \tfrac{1}{N_z} \textstyle\sum_{l=1}^{N_z} H_l$$

Fig. 3.5 shows a Monte Carlo approximation of the entropy of the robot pose, conditioned on a single future observation set, in a simulated 2-D environment. The results hint at several important considerations regarding active localization. In particular, there is a correlation among the landmark distribution in the environment, the sensing range and field of view of the robot, and the length of the planning horizon T that affects the performance. On one hand, if the sensing range and the field of view are unrestricted, there would be no need to for active localization. The filtering process alone will be able to uniquely identify the robot pose. On the other hand, since the planning process is inherently local, if the horizon T is not long enough to reach perceptually-distinct areas in the environment, the robot can get stuck in a local maximum (the flat red region in Fig. 3.5) of the entropy surface. Then, all considered motions will have the same cost and no progress will be made. Active localization becomes particularly attractive when the sensing range and the field of view are limited but the environment contains distinct landmarks within the reachable (in T steps) sensing perimeter. In such scenarios, the planning process can improve both the efficiency and accuracy of the localization filter.

Localization as a Secondary Objective

Often times, localization is a requisite but secondary objective for a mobile robot. A robot typically needs to avoid collisions and reach a primary objective, such as a goal pose in the environment. As discussed by Fox et al. (1998), an additional term can be added to the cost function in order to minimize the probability that a control sequence leads to a collision. Dealing with obstacles in the environment correctly also requires that object visibility is accounted for both in the probability of detection $p_d(y, x)$ and in the sampling of measurement sets for the Monte Carlo evaluation of the conditional entropy. Once the robot is localized well, it can plan a global path $\mathcal{P} = \{\rho\}$, consisting of a sequence of poses ρ, which leads the robot to its ultimate goal. Along the way, if re-localization is necessary the robot should not deviate significantly from the intended path $\mathcal{P}$. Thus, we consider the following three-fold objective:

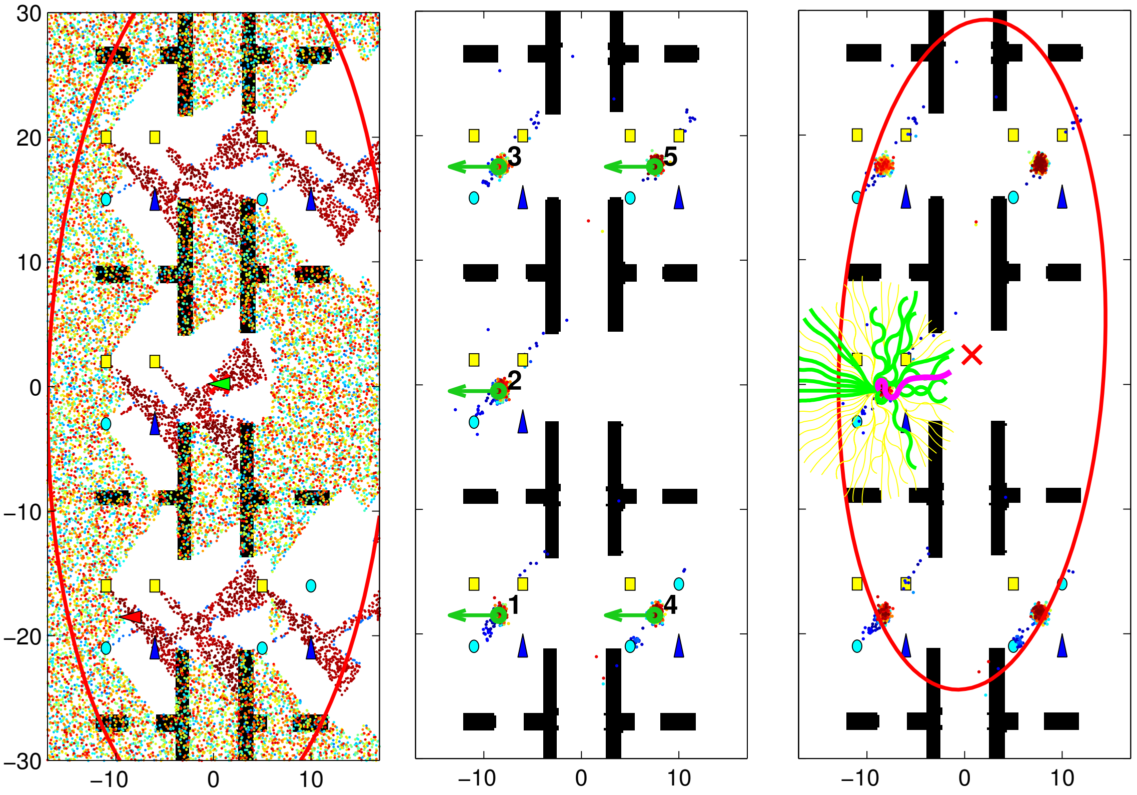
$$\begin{aligned} \sigma^* \in \operatorname*{arg\,min}_{\sigma \in \Sigma} \;\; & \alpha_1 \mathbf{H}(x_{0:T} \mid Z_{1:T}) + \alpha_2 \mathbb{E}\left[\min_{\rho \in \mathcal{P}} d(x_T, \rho)\right] + \alpha_3 \max_{s=1}^{T} \mathbb{P}(x_s \in \text{Collision}) \\ \text{s.t.} \;\; & x_{s+1} \sim p_f(\cdot \mid x_s, \sigma_s), \;\; s = 0, \ldots, T-1 \\ & Z_s \sim p(\cdot \mid Y_d(x_s), x_s), \;\; s = 1, \ldots, T \end{aligned}$$

where $\mathbb{E}\left[\min_{\rho \in \mathcal{P}} d(x_T, \rho)\right]$ is the expected minimum deviation of the final pose x_T from the global path $\mathcal{P}$ and $\max_{s=1}^T \mathbb{P}(x_s \in \text{Collision})$ is the maximum probability of collision along the chosen trajectory. The constants $\alpha_1, \alpha_2, \alpha_3$ specify the relative importance of the three objectives. Due to the "perfect motion" assumption, the last two terms in the cost function can be computed as follows:

$$\begin{aligned} \max_{s=1}^{T} \mathbb{P}(x_s \in \text{Collision}) &= \max_{s=1}^{T} \left(\sum_{k=1}^{N} \mathbb{1}\{x_{s|s}^k \in \text{Collision}\} w_{0|0}^k \right) \\ \mathbb{E}\left[\min_{\rho \in \mathcal{P}} d(x_T, \rho) \right] &= \sum_{k=1}^{N} \left(\min_{\rho \in \mathcal{P}} d(x_{T|T}^k, \rho) \right) w_{0|0}^k, \end{aligned} \tag{3.18}$$

where $\mathbb{1}\{x_{s|s}^k \in \text{Collision}\}$ is the indicator of the set $\{x_{s|s}^k \in \text{Collision}\}$.

The performance of the active semantic localization approach is demonstrated in simulation with a differential-drive robot in Fig. 3.6 and Fig. 3.7. The task of the robot is to localize itself and subsequently reach a goal pose specified on the prior map. The initial particle set is uniformly distributed over the whole environment. As a result, minimizing the entropy in the early iterations will be expensive and of little value. In our experiments, the robot either acquires several observation sets without moving (as in Fig. 3.6 (A)) or chooses motion primitives which minimize the collision probability only (by setting $\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$). Once the summarized particle set contains less hypotheses, both the entropy and the probability-of-collision criteria can be enabled to select informative trajectories (see Fig. 3.6 (B) for details). We used $\alpha_1 = 0.55, \alpha_2 = 0, \alpha_3 = 0.45$ before the first time the robot is localized well (the covariance of the particle set is small). Once well-localized, the robot can plan a path from the mode of the distribution to the goal pose. In our experiments, we used A^* with a cost map that rewards landmark visibility. If along the way to the goal the uncertainty in the robot pose starts to increase due to the motion noise, the robot can carry out the minimization in (3.18) with all three terms enabled. We used $\alpha_1 = 0.5, \alpha_2 = 0.05, \alpha_3 = 0.45$ in this case (see Fig. 3.7 (B) for details). The experiments demonstrate that the robot achieves global localization autonomously, avoids collisions in the environment, re-localizes itself if necessary, and reaches the goal successfully. Additional simulations, which compare our approach to other active localization techniques, are presented in Sec. 3.1.6.

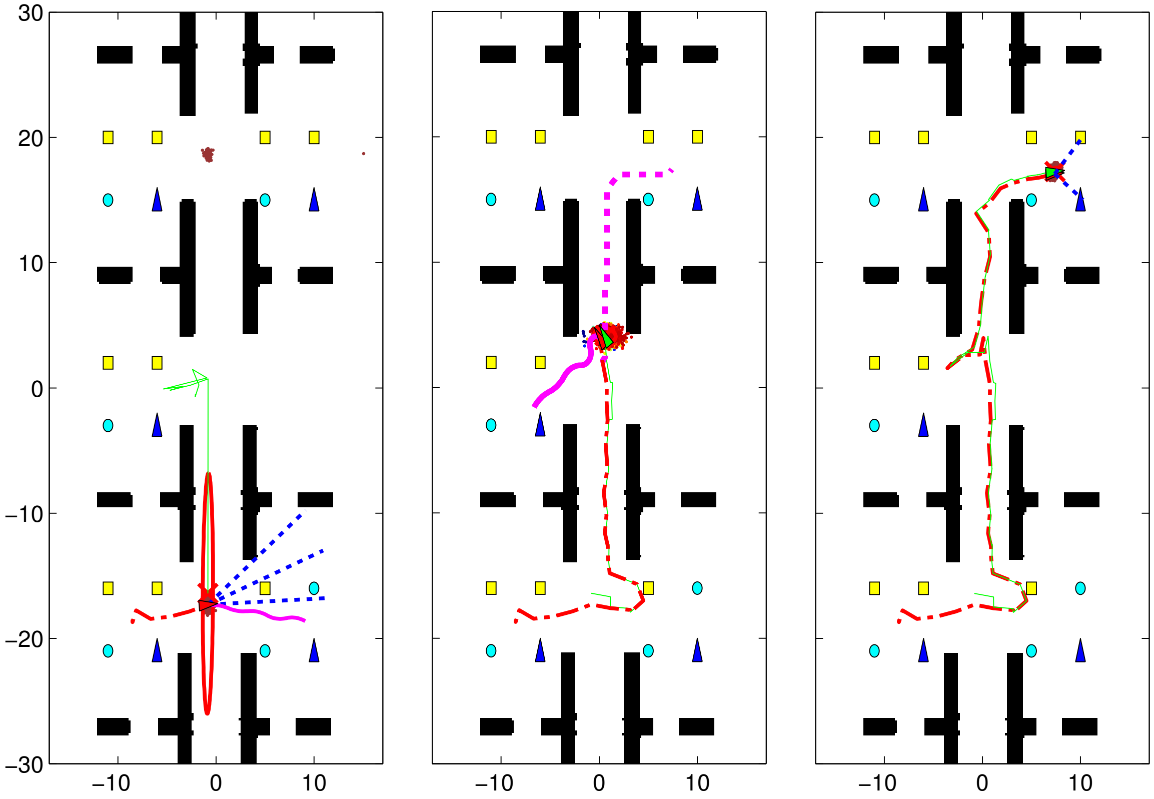


(a) Particle set after the first set of semantic observations, particle mean (green triangle), particle covariance (red ellipse), and the actual unknown robot pose (red triangle)

(b) The particle distribution converges to 5 ambiguous poses (green arrows) after several semantic observations. Pose 4, unlike the rest, has two cyan circles in its vicinity.

(c) All motion primitives (yellow, in frame of reference of pose 2), minimum-collision-probability motion primitives (green), and minimum-entropy-and-collision-probability motion primitive (magenta)

Figure 3.6: A simulation of a differential-drive robot employing our active semantic localization approach to reach a goal. The environment contains objects from three classes (square, circle, and triangle) in six areas, divided by the black obstacles. The task of the robot is to localize itself (position and orientation) and reach pose 5, indicated by the green arrow on subplot (b). It has a field of view of 94° and a sensing range of 12.5m. The other parameters of the observation model were $p_0 = 0.73, m_0 = 2.7, v_0 = 35, \Sigma_\beta = 5°, \lambda = 0.5$. The robot had no prior information about its initial pose (subplot (A)). The particle distribution converges to 5 ambiguous locations after several semantic observations because a yellow square and a cyan circle are detected repeatedly (subplot (B)). The robot plans its motion (using the motion primitives in Fig. 3.4) to minimize the probability of collision and the entropy of its pose, conditioned on 5 future sets of semantic observations (subplot (C)). The description continues in Fig. 3.7.



(a) Actual robot trajectory (dotted red), estimated trajectory (green), particle distribution (dark red), particle covariance (red ellipse), and observation set (dotted blue lines) after the second cyan circle in area 4 (bottom right) is observed

(b) Actual (dotted red) and estimated (green) robot trajectories, intended trajectory towards the goal (dotted magenta), and minimum-entropy-and-collision-probability motion primitive (solid magenta) chosen to decrease the pose uncertainty

(c) The robot managed to re-localize itself and reach the goal successfully

Figure 3.7: Continuation of the active semantic localization simulation from Fig. 3.6. The robot recognizes correctly that the best way to disambiguate its pose is to visit the bottom-right area (subplot (A)). At this point, there are only two remaining hypotheses and more weight is starting to concentrate around the true pose. Once the robot considers itself localized (the covariance of the particle set is small), it plans a path to the goal in the top-right area. As there are no landmarks along the hallway, the motion noise causes the uncertainty in the robot pose to increase. Using the entropy-minimization criterion, the robot recognizes that it needs to deviate from its intended path and visit an area with landmarks in order to re-localize (subplot (B)). The robot reaches the goal successfully (subplot (C)).

3.1.6 Performance Evaluation

This section evaluates the performance of the semantic localization approach in simulation and in three real-world scenarios. Global localization from semantic observations is demonstrated for a differential-drive robot, for a Tango phone (Google ATAP group 2014), and on the KITTI visual odometry dataset (Geiger et al. 2013).

Semantic information was obtained using deformable part models (DPM, Felzenszwalb et al. 2010b), which achieve excellent performance in single-image object recognition. Given an input image, an image pyramid is obtained via repeated smoothing and subsampling. Histograms of oriented gradients are computed on a dense grid at each level of the pyramid. For each object class (in the set $\mathcal{C}$), a detector is applied in a sliding-window fashion to the image pyramid, in order to obtain detection scores at each pixel and scale (see Fig. 3.8). The detection scores above a certain threshold are returned, along with bounding box, class, and bearing information. The collection of all such measurements at time t forms the random finite set Z_t. A significant increase in detection speed is obtained via an active DPM approach, which optimizes the order of filter evaluations and the time at which to stop and make a decision. The active DPM approach will be discussed in more detail in Ch. 4, Sec. 4.5.

In the experiments, it was sufficient to represent the state of an object y with its position and class because orientation and appearance variations are captured well by a DPM-based detector. If necessary, our model can incorporate appearance and shape signatures by extending the object state y and training an appropriate observation model. This is likely to make the data association more unimodal (i.e., make the terms in the sum in (3.9) dominated by the weight of a single data association), in which case the maximum likelihood data association approach (Sec. 3.1.5) would perform well. We emphasize that the permanent approach can handle this scenario efficiently too. As permanent approximation methods rely on Monte-Carlo sampling from the data associations, fewer samples can be used to speed up the computations. The connection between the observation model and the permanent incorporates this naturally and leverages state-of-the-art algorithms.

Mobile Robot Semantic Localization

We carried out simulations and real-world experiments in an indoor environment using a differential-drive robot equipped with an inertial measurement unit (IMU), magnetic wheel encoders, a Kinect RGB-D camera with Nyko Zoom wide-angle lens, and a Hokuyo UTM-30LX 2D laser range finder. The IMU and the encoders were integrated using a differential-drive model (Appendix B.1) and Gaussian noise was added to obtain the motion model in (3.1). Only the RGB images were used for semantic observations via Alg. 7. The measurement updates were performed using the exact permanent algorithm (Alg. 6). The lidar was used to provide a ground-truth trajectory in the real-world experiments via geometric Monte-Carlo localization. The performance is demonstrated for *global localization*, which means that the robot had no prior information about its initial pose.

The detection model $p_d(y, x)$, the measurement likelihood $p_z(z \mid y, x)$, and the clutter model $p_\kappa(z)$ were obtained from training data as discussed in Sec. 3.1.1. The angle of view of the wide-angle lens was 94°, the detection range was 10 meters, and the following parameters were learned: $p_0 = 0.92, m_0 = 3.5, v_0 = 20.52, \Sigma_\beta = 4^\circ$. The semantic map

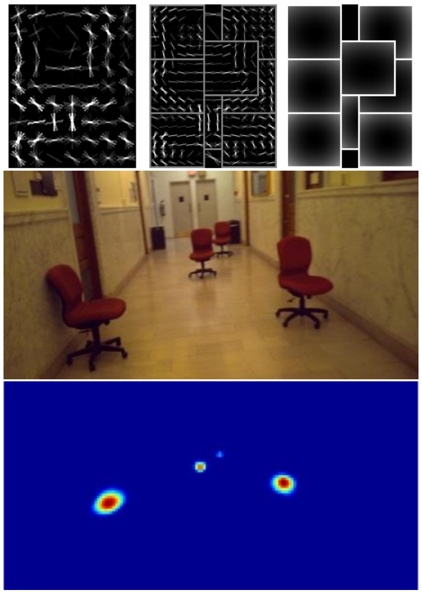


Figure 3.8: A component of the deformable part model of a chair (top) and scores (bottom) from its evaluation on an image (middle) containing four chairs

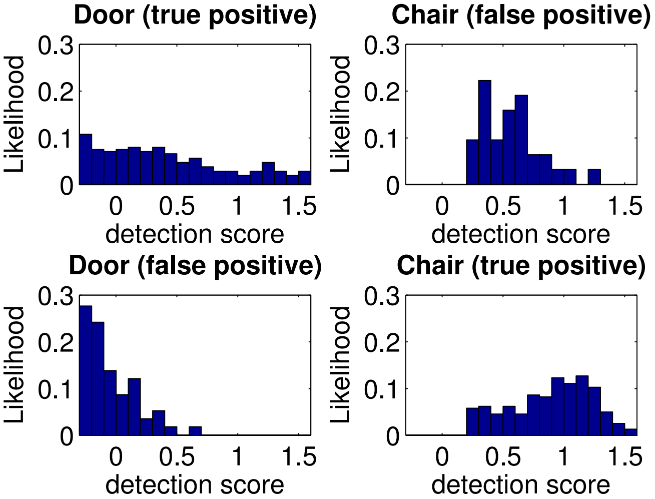


Figure 3.9: Detection score likelihoods obtained from training images

in the real-world experiment contained doors (class 1) and chairs (class 2). The confusion matrix was:

$$p_c(c \mid y^c) = \begin{bmatrix} 0.94 & 0.08 \\ 0.06 & 0.92 \end{bmatrix}$$

while the single-object observation models are shown in Fig. 3.9.

A simulated environment of size 25×25 m^2 was populated by objects with randomly-chosen positions and classes (see Fig. 3.10). The robot motion was simulated along a pre-specified trajectory. Semantic observations were simulated using the learned detection, clutter, and measurement likelihood models. The error in the estimates, averaged over 50 repetitions with different randomly-generated scenes, is presented in Fig. 3.13. Since the robot starts with a uniform prior over the whole environment, the error is large in the initial iterations. Multiple hypotheses are present until enough semantic measurements are obtained to localize the robot uniquely. The performance is also demonstrated in a challenging scenario with a lot of ambiguity in Fig. 3.11. The reason for using only two classes in the experiments was to increase the ambiguity in the data association. Our approach can certainly handle more classes and a higher object density. Fig. 3.12 shows a simulation with clutter rate $\lambda = 4$ and 150 objects from 5 classes in a 25×25 m^2 area. Scenarios with such high object density necessitate the use of an approximate permanent algorithm for real-time operation.

In the real experiments, the robot was driven through a long hallway containing doors and chairs. Four data sets were collected from the IMU (at 100 Hz), the encoders (at 40 Hz), the lidar (at 40 Hz), and the RGB camera (at 1 Hz). Lidar-based geometric localization was performed via the ROS amcl package (Howard and Gerkey 2002) and the results were used as the ground truth. Appendix D.16 Extension 2 contains a video of the experiment.

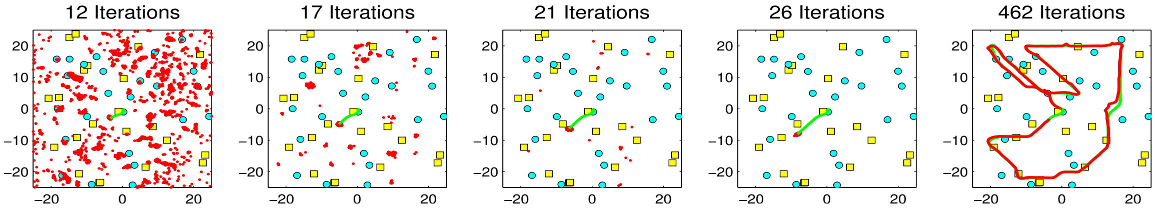


Figure 3.10: A simulated environment with 45 objects from two classes (yellow squares, blue circles). The plots show the evolution of the particles (red dots), the ground truth trajectory (green), and the estimated trajectory (red). The expected number of clutter detections was set to $\lambda = 2$.

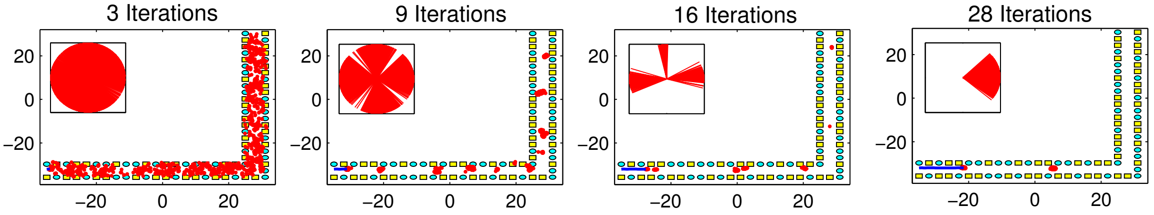


Figure 3.11: A simulated example of semantic localization in the presence of severe perceptual aliasing. The ground truth trajectory (blue) and the evolution of the particle positions (red points) and orientations (red lines, top left) are shown.

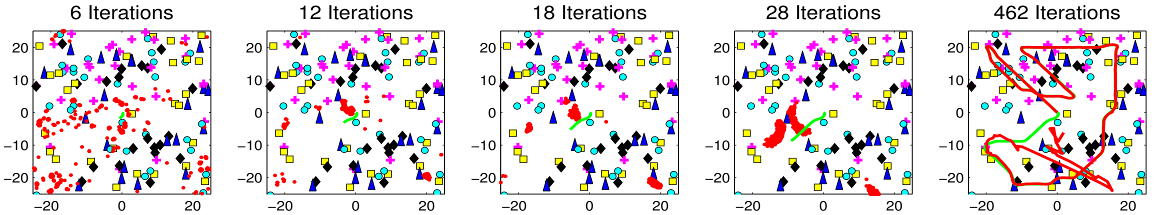
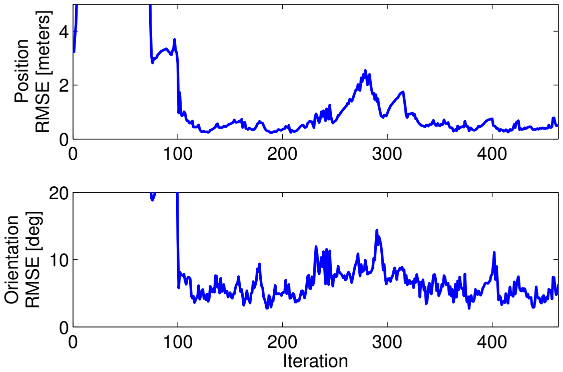


Figure 3.12: A simulated environment with 150 objects from 5 classes (circles, squares, triangles, crosses, and diamonds) in a 25 × 25 m^2 area. The plots show the particles (red dots), the ground truth trajectory (green), and the estimated trajectory (red) for clutter rate $\lambda = 4$.



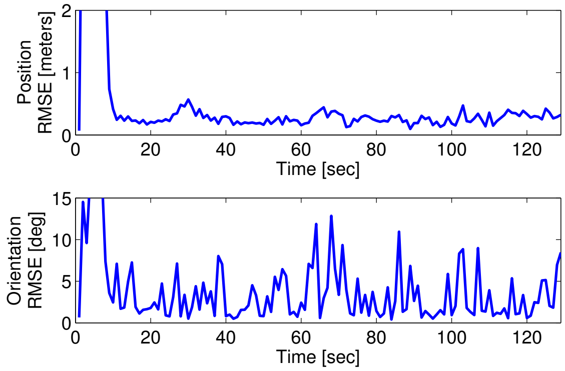


Figure 3.13: Root mean squared error (RMSE) in the pose estimates obtained from the semantic localization algorithm after 50 simulated runs of the scenario in Fig. 3.10

Figure 3.14: Root mean squared error (RMSE) between the pose estimates from semantic localization and from lidar-based geometric localization obtained from four real experiments

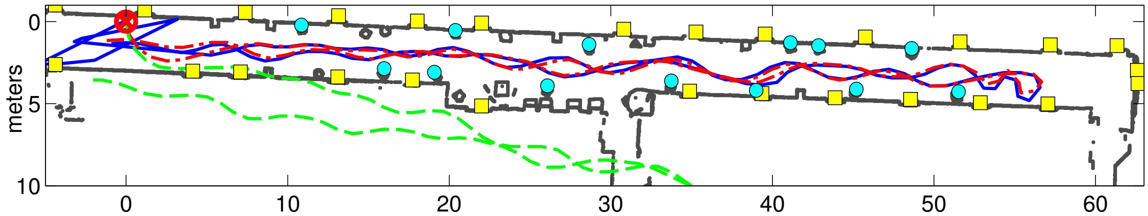


Figure 3.15: Robot trajectories estimated by lidar-based geometric localization (red), image-based semantic localization (blue), and odometry (green) in a real experiment. The starting position, the door locations, and the chair locations are denoted by the red cross, the yellow squares, and the cyan circles, respectively. See Appendix D.16 Extension 2 for more details.

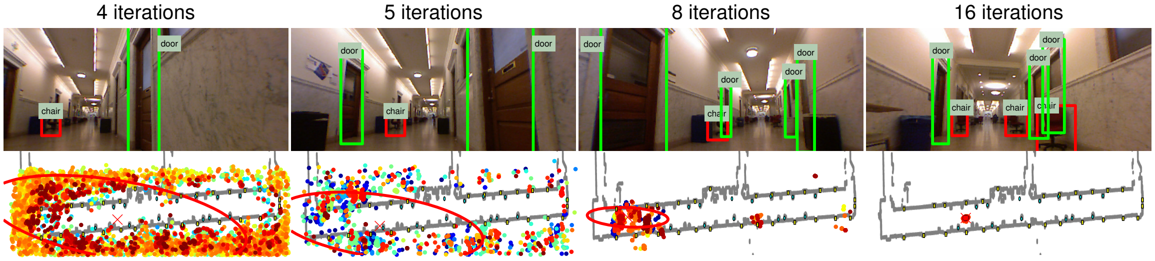


Figure 3.16: Particle filter evolution (bottom) and object detections (top) during a real semantic localization experiment

The lidar and semantic estimates of the robot trajectory are shown in Fig. 3.15. The error between the two, averaged over the 4 runs, is presented in Fig. 3.14. The error is large initially because, unlike the lidar-based localization, our method was started with an uninformative prior. Nevertheless, after the initial global localization stage, our approach achieves average errors in the position and orientation estimates of less than 35 cm and 10°, respectively. The particle filter evolution is illustrated in Fig. 3.16 along with some object detections.

Comparison with Maximum Likelihood Data Association

We compared the permanent-based data association (PER) to the more traditional maximum likelihood data association (MLD), used for example in FastSLAM (Montemerlo and Thrun 2003). PER is based on Alg. 7 with an exact permanent computation (Alg. 6) on line 6. MLD is based on Alg. 7 also but the set of detections on line 6 is processed sequentially as described in Sec. 3.1.5. The two approaches were compared on the four real datasets (Fig. 3.15) and on the simulations in Fig. 3.10 and Fig. 3.11. Because we assume semantically-meaningful measurements, the observation sets in our comparison had relatively low cardinalities. Of course, if there are many observations per time step (e.g., SIFT features), MLD would be significantly more efficient than the exact permanent algorithm. In future work, we plan to compare MLD to an approximate permanent algorithm.

The comparison is presented in Table 3.1 for two types of initializations: local (L), for which the initial particle set had errors of up to 1 m and 30°, and global (G), for which the set was distributed uniformly over the environment. MLD(L) performs as well as PER(L) in the real experiments and in Fig. 3.11. In Fig. 3.10, the data association is highly multi-modal and MLD(L) does not converge even with 15K particles. This is reinforced in

Table 3.1: Comparison of maximum likelihood data association (MLD) and our permanent-based data association approach (PER) with the exact permanent computation (Alg. 6) on the four robot datasets (Fig. 3.15) and the simulations in Fig. 3.10 and Fig. 3.11. Two types of initializations were used: local (L), for which the initial particle set had errors of up to 1 m and 30°, and global (G), for which the initial particle set was uniformly distributed over the whole environment. Number of particles (NP) in thousands, position error (PE), orientation error (OE), and filter update time[8](UT), averaged over time, are presented. The first MLD(G) column uses the same number of particles as PER(G), while the second uses a large number in an attempt to improve the performance.

Fig. 3.15	MLD(L)	MLD(G)	MLD(G)	PER(L)	PER(G)
NP [K]	0.50	3.00	40.0	0.50	3.00
PE [m]	0.26	22.9	0.31	0.26	0.26
OE [deg]	2.54	107	2.75	2.67	2.69
UT [sec]	0.023	0.060	0.600	0.065	0.320
Fig. 3.10					
NP [K]	0.50	5.00	100	0.50	5.00
PE [m]	15.3	24.9	17.3	0.32	0.72
OE [deg]	67.0	68.8	72.8	4.58	9.17
UT [sec]	0.012	0.062	1.100	0.042	0.400
Fig. 3.11					
NP [K]	0.50	24.0	100	0.50	24.0
PE [m]	0.27	48.8	26.9	0.11	2.35
OE [deg]	3.68	112	74.9	2.08	4.05
UT [sec]	0.027	0.760	3.340	0.062	2.620

the global initialization cases. While PER(G) performs well with 3K particles, MLD(G) needs 40K to converge consistently on the real datasets and is slower at the same level of robustness. In Fig. 3.10 and Fig. 3.11, MLD(G) does not converge even with 100K particles. We conclude that once the particles have converged correctly MLD performs as well as PER. However, with global initialization or ambiguous data association, MLD makes mistakes and can never recover, while PER is robust with a small number of particles.

Global Localization for Project Tango

The Project Tango phone (Google ATAP group 2014) is designed to track its full 3-D motion and create a geometric map of the environment. However, it does not use an absolute reference to the world and provides localization only with respect to its initial frame of reference. We demonstrate that our semantic localization approach can provide global positioning of the Tango phone within an existing map of semantically-meaningful landmarks.

The Tango phone is equipped with an IMU, a front-facing (120° field of view) camera, a rear-facing RGB/IR narrow (68° field of view) camera, and a rear-facing fisheye (180° field of view) camera. It provides a 6-D position-quaternion estimate of its pose and associated covariance, over time, within the initial frame of reference. In our experiments, this local trajectory was used as the motion model (3.1) in the prediction step in Alg. 7. The update step was performed using semantic observations (class, score, and bearing) only from the narrow camera RGB images. The same hallway as in the robot experiment was

[8] The reported times are from a MATLAB implementation on a computer with i7 CPU@2.3GHz and 16GB RAM. The timing results include only the time needed to perform data association and update the weights for all particles. The time required for object recognition is not included because it is the same for both methods.

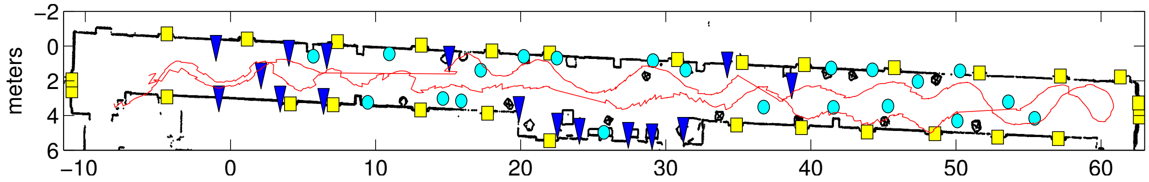


Figure 3.17: Tango phone trajectory (red) estimated via semantic localization in a real experiment. The semantic map contains doors (yellow squares), red chairs (cyan circles), and brown chairs (blue triangles). Ground-truth information was not available for this experiment. See Appendix D.16 Extension 4 for more details.

traversed several times with the phone. RGB images from the narrow camera (at 30 Hz) and the Tango visual odometry (at 30 Hz) were recorded. The prior semantic map of the environment (see Fig. 3.17) contained doors (class 1), red chairs (class 2), and brown chairs (class 3). Two of the runs were used to train the object detector and to learn the observation model parameters: sensing range 15 m, $p_0 = \begin{pmatrix}0.71 & 0.81 & 0.82\end{pmatrix}^T$, $v_0 = 35.4$, $m_0 = 2.7$, $\Sigma_\beta = 5°$, $\lambda = 0.76$, and confusion matrix:

$$p_c(c \mid y^c) = \begin{bmatrix}0.98 & 0 & 0\\ 0 & 0.94 & 0.08\\ 0.02 & 0.06 & 0.92\end{bmatrix}.$$

The semantic localization approach was used to recover the global Tango trajectories in the rest of the runs. Since the prior semantic map contained 2-D object positions, only the horizontal bearing angle was used to update the 2-D position and yaw angle of the phone. A good estimate of the phone's pitch and roll angles can be obtained from the local 6-D trajectory (provided by the Tango phone). Thus, the global semantic localization was performed in 5-D (without the z-axis). This can be extended, of course, if vertical bearing measurements are used and the landmarks in the prior map are annotated with z-coordinates. The likelihoods of the semantic observations were computed via the exact permanent algorithm (Alg. 6). Videos, from two of the experiments, are provided in Appendix D.16 Extension 3 and Extension 4. The phone trajectory, recovered from the second run (Extension 4), is shown in Fig. 3.17. Unfortunately, ground-truth trajectories are not available for these experiments. The videos show 9 global localization trials, in which, on average, 11 sets of semantic observations were needed to obtain an accurate estimate of the phone pose in the prior map. They demonstrate that our algorithm can *repeatedly relocalize* and track the phone pose within the same environment. Moreover, our semantic localization approach is very robust to perceptual aliasing and can improve the visual odometry provided by the phone in ambiguous environments and when closing loops.

Evaluation on the KITTI Dataset

This section evaluates the performance of the semantic localization approach on the KITTI visual odometry dataset (Geiger et al. 2013). The dataset consists of 22 sequences of color stereo images (0.5 Megapixels in png format) and 3-D Velodyne point clouds (100k points per frame) recorded from a vehicle, driving through residential areas. Eleven sequences (00-10) contain ground-truth vehicle trajectories provided by a GPS/IMU system with real-time kinematic correction. Only sequences $\{00, 05, 06, 07, 08, 09, 10\}$ were used in our experiments

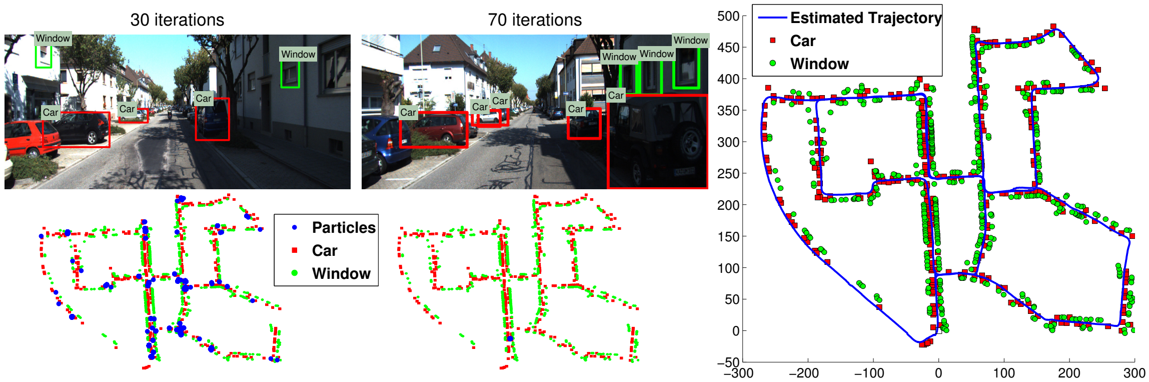


Figure 3.18: Vehicle trajectory estimated via global semantic localization on sequence 00 from the KITTI visual odometry dataset. The left and middle plots show two images with car and window detections and the corresponding particle distributions in the semantic map. The plot on the right shows the semantic map and the trajectory, recovered after unique localization (iteration 70). See Appendix D.16 Extensions 5 and 6 for more details.

because the rest either had too few static landmarks or did not contain ground-truth information. The cars (class 1) and windows (class 2) in the RGB image sequences were labeled in order to build prior semantic maps. The Velodyne range information was mapped to the images and the ground-truth trajectories were used to project the labeled objects to the 3-D world coordinate frame. The final semantic maps are provided in Appendix D.16 Extension 14 and the map for sequence 00 is shown in Fig. 3.18.

The pre-trained deformable-part car models provided in the KITTI object dataset were used for car recognition. Sequence 07 was used to train a deformable-part-model-based window detector and to obtain parameters for the single-object detection model $p_d(y, x)$, observation likelihood $p_z(z \mid y, x)$, and clutter model $p_\kappa(z)$. The car detection model was nonzero for a distance range of $[3, 33]$ meters and used parameters $p_0 = 0.7, m_0 = 11.8, v_0 = 14$. The window detection model was nonzero in the range $[7, 24]$ with parameters $p_0 = 0.7, m_0 = 12.7, v_0 = 7$. The rest of the parameters were: $p_c(c \mid y^c) = I_2$, sensing range 32 m, field of view 80°, $\Sigma_\beta = 5^\circ$, and $\lambda = 0.5$.

Visual odometry via Viso2 (Geiger et al. 2011) was used for the prediction step in Alg. 7. Viso2 provides a 6-D local trajectory. As in the Tango phone experiments, only the 2-D position and the yaw angle of the trajectory were updated via our method. The likelihoods of the car-window semantic observations were computed via the exact permanent algorithm (Alg. 6). Global semantic localization was carried out on sequences 00, 05-10. The vehicle trajectory, recovered from sequence 00, is shown in Fig. 3.18 along with some object detections. A video of the experiment is included in Appendix D.16 Extension 6. An additional experiment, in which the localization was restarted every 400 iterations, was carried out on sequence 00 and is presented in Appendix D.16 Extension 5. The experiment demonstrates that our algorithm can repeatedly and successfully relocalize and track the vehicle pose within the same environment. Finally, the results of the global semantic localization on the rest of the sequences (05-10) are presented in Appendix D.16 Extensions 7 - 12. The localization errors with respect to the ground-truth trajectories from all experiments are presented in Fig. 3.19. Initially, the errors are large because our method is started with an uninformative prior (a uniform distribution in the area around the

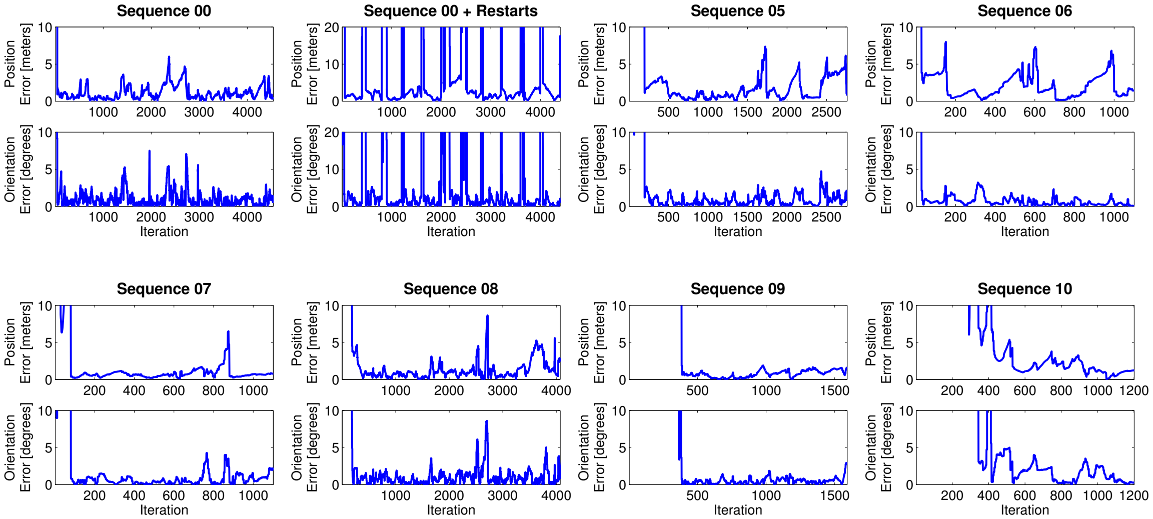


Figure 3.19: Position (euclidean distance) and orientation errors of the vehicle trajectories recovered via global semantic localization on sequences $\{00, 05, 06, 07, 08, 09, 10\}$ from the KITTI visual odometry dataset. The plot, titled "Sequence 00 + Restarts", shows results from an experiment in which the localization was restarted every 400 iterations. Appendix D.16 Extensions 5 - 13 provide videos of all experiments.

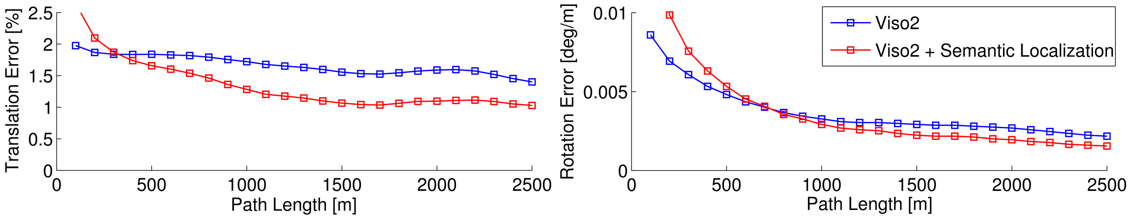


Figure 3.20: Average translation and orientation errors[8]obtained from visual odometry via Viso2 (Geiger et al. 2011) and from visual odometry combined with semantic localization on sequences $\{00, 05, 06, 07, 08, 09, 10\}$ from the KITTI visual odometry dataset. Both methods use a known starting vehicle pose, i.e., perform tracking instead of global localization.

landmarks). Nevertheless, after the initial global localization stage, our approach achieves average errors in the position and orientation estimates of less than 1 m and 5°, respectively. Even though the data association obtained from permanent computations is very robust to perceptual aliasing, sometimes, the ambiguity in the environment is large enough to cause particle depletion problems. For example, if resampling is done too frequently and there is no way to detect if the system is lost (i.e., the particle distribution is never reset), the localization might fail. Such a fail case is shown in Appendix D.16 Extension 13.

To demonstrate that localization from semantic observations is complementary to existing odometry and SLAM techniques, we also carried out tracking experiments, in which the initial vehicle pose was known. In Fig. 3.20, the position and orientation errors obtained from visual odometry are compared to those obtained from visual odometry, combined with our semantic localization approach. Even though visual odometry provides excellent tracking results by itself, the addition of semantic observations provides a reference to the absolute (semantic map) frame and improves the results.

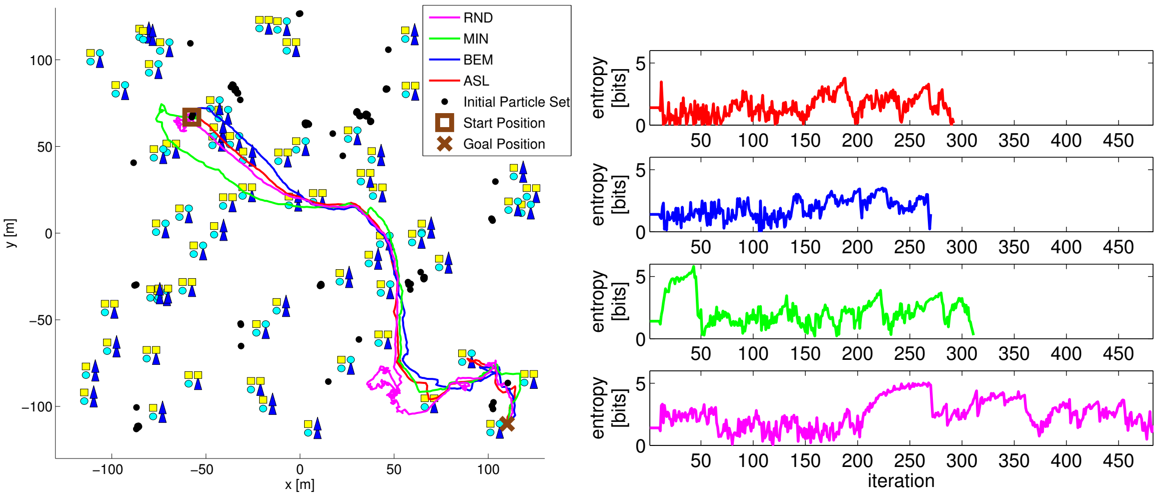


Figure 3.21: The left plot shows the trajectories, followed by four different active-semantic-localization approaches, which localize and lead a differential-drive robot to a goal pose in a simulated environment containing 300 objects from 3 classes (yellow square, cyan circle, blue triangle). The initial particle distribution is shown by the black dots. The four methods are: (1) ASL: active semantic localization presented in Sec. 3.1.5, (2) RND: chooses motion primitives at random, (3) MIN: chooses the motion primitive that drives the particle mean closest to the closest landmark, (4) BEM: bearing-only entropy minimization (see text for details). The right plot shows the particle-distribution entropies along the trajectories associated with each method.

Active Semantic Localization Simulations

In this section, we evaluate the performance of the active semantic localization approach (Sec. 3.1.5) by comparing it to three other active localization methods in simulation. The simulations use the differential-drive robot model and the motion primitives in Fig. 3.4. Fifty environments of size $120 \times 120 \; m^2$ containing 300 objects from 3 classes (square, circle, triangle) were generated by sampling random points and placing one of three possible four-object structures (square-square-circle-triangle, square-circle-circle-triangle, or square-triangle-circle-triangle) in order to create perceptual ambiguity (see Fig. 3.21). A start and a goal pose for the robot were chosen in the top-left and bottom-right corners of each environment, respectively. The semantic localization algorithm (Alg. 7 with exact permanent computation via Alg. 6) was initialized with a uniform particle distribution over the whole environment. Semantic observations were simulated using the learned detection, clutter, and measurement likelihood models for cars and windows (from the KITTI dataset experiments) and red chairs (from the Tango phone experiments), corresponding to squares, circles, and triangles, respectively. For each environment, semantic observations were collected, while moving in a straight line, and the filter was updated without resampling until there were 100 effective particles (see Fig. 3.21 for an example of the initial particle set). Starting with this initial particle distribution, the following methods were compared to our active semantic localization (ASL) approach:

- RND: chooses a motion primitive from the ones in Fig. 3.4 at random

[8] The orientation error is the cumulative error in degrees between the ground-truth and the estimated orientations divided by path length in meters.

Table 3.2: Comparison of the average (over 50 simulated environments) performance of the four active-semantic-localization approaches, referenced in Fig. 3.21. The average euclidean distance between the start and the goal positions was 251 m. If the goal was not reached in 1000 iterations, the experiment was terminated. The table presents averages of the number of iterations until termination, the euclidean distance to the goal at termination, the entropy in the particle distributions, and the position and orientation errors with respect to the ground-truth robot trajectory.

Method	Steps	Distance to Goal [m]	Entropy [bits]	Position Error [m]	Orientation Error [deg]
RND	595	35.02	3.00	22.17	12.48
MIN	508	35.58	2.83	27.58	11.98
BEM	447	16.22	3.05	18.49	11.13
ASL	345	12.09	2.05	12.59	6.37

- MIN: chooses the motion primitive, which drives the mean of the particle distribution closest to the closest landmark in the environment

- BEM: chooses a motion primitive by minimizing the entropy of the robot pose conditioned on the future bearing measurements only (see Appendix D.14 for details).

The methods were used to choose motion primitives if the entropy in the particle distribution (computed by discretizing the robot state space into cells of size $2.5 \times 2.5\ m^2$ and 25° and replacing the particles, contained in the same cell, with their average) increased above 2.5 bits. Otherwise, if the entropy decreased under 2.5 bits, each method planned a trajectory from the mean of the particle distribution to the goal state using A* and followed it using a deterministic controller. The trajectories, followed by the four active localization methods, and the associated particle-distribution entropies are shown for one of the simulated environments in Fig. 3.21. Performance statistics, averaged over the 50 environments, are presented in Tab. 3.2. The results show that, on average, the active semantic localization approach reaches the goal in less iterations, with lower particle-distribution entropy, and with lower estimation error, compared to the other three approaches. Of course the approach is also more computationally demanding. As expected, the random approach performs the worst because when there no landmarks in proximity to the robot, it might spend a long time until relocalization. The BEM approach demonstrates much better performance but the main problem with it and the MIN method is that they rely on the mean of the particle distribution for planning. Sometimes, when the mean is far from the true pose, these methods choose trajectories, which do not necessarily result in improved localization accuracy.

3.2 Model-free Source Seeking

In this section, we switch gears and address a more extreme scenario in which the observation model is completely unknown. As mentioned in the introduction, we consider a somewhat simpler objective - to control a sensor team in order to localize a *single static* source that generates the sensed signal. Suppose that the team is composed of n sensing robots with states $\{x_{1,t}, \ldots, x_{n,t}\} \subset \mathcal{X} \cong \mathbb{R}^{d_x}$ at time t. The states are typically comprised of pose and velocity information but might include other operational parameters too. At a high-level planning stage we suppose that the vehicles have discrete single-integrator dynamics $x_{i,t+1} = x_{i,t} + u_{i,t}$, where $u_{i,t} \in \mathcal{U}$ is the control input to sensor i. The task is to localize a signal source, whose state is $y \in \mathcal{Y} \cong \mathbb{R}^{d_y}$ captures the source position and other observable properties of interest. At time t, each sensor i has access to a noisy measurement $z_{i,t} \in \mathcal{Z} \cong \mathbb{R}^{d_z}$ of the signal generated by y:

$$z_{i,t} = h(x_{i,t}, y) + v_{i,t}, \tag{3.19}$$

where $v_{i,t}$ is the measurement noise, whose values are independent at any pair of times and among sensors. The noise depends on the states of the sensor and the source but to simplify notation we do not make it explicit. We assume that the noise is zero-mean and has a finite second moment, i.e., $\mathbb{E}v_{i,t} = 0,\ \forall i, t, x_{i,t}$ and $\operatorname{tr}\big(\mathbb{E}[v_{i,t}v_{i,t}^T]\big) < \infty$. In the reminder, we use the notation $x_t := \big[x_{1,t}^T, \ldots, x_{n,t}^T\big]^T$, $u_t := \big[u_{1,t}^T, \ldots, u_{n,t}^T\big]^T$, $z_t := \big[z_{1,t}^T, \ldots, z_{n,t}^T\big]^T$, and $v_t := \big[v_{1,t}^T, \ldots, v_{n,t}^T\big]^T$.

In the model-free scenario, the sensors simply receive measurements *without knowing the signal model* $h(\cdot,\cdot)$. We suppose that the team adopts some arbitrary formation, with center of mass $m_t := \sum_{i=1}^n x_{i,t}/n$ at time t, which can be enforced using potential fields (Ögren et al. 2004) or convex optimization (Derenick and Spletzer 2007). The sensors use the centroid m_t as the estimate of the source state y at time t and try to lead it toward the true source location based on the received measurements. Let $\pi : \mathcal{X} \to \mathcal{Y}$ be a known transformation which maps the team centroid to a source estimate. For example, if the robot state space captures both position and orientation, e.g., $\mathcal{X} = SE(2)$, but we are interested only in position estimates for the source, e.g., $\mathcal{Y} = \mathbb{R}^2$, then π will be the projection that extracts the position components from the centroid $m_t \in \mathcal{X}$. We consider the following problem.

Problem (Model-free Source Seeking)**.** *Assume that the measurement signal in (3.19) is scalar*[9] *and its expectation is maximized at the true state y of the source:*

$$\pi^{-1}(y) \in \arg\max_{x \in \mathcal{X}} h(x, y). \tag{3.20}$$

Generate a sequence of control inputs $u_0, u_1, \ldots$ for the team of sensors in order to drive its centroid m_t toward a maximum of the signal field $h(\cdot, y)$.

We resort to a greedy control approach, based on stochastic gradient descent, that results in a simple algorithm with local convergence guarantees and is applicable to robot platforms with limited computational resources and no global localization capabilities.

3.2.1 Stochastic Finite-difference Gradient Ascent

We design an iterative optimization scheme which causes the centroid m_t of the robot formation to ascend the gradient $g(x, y) := \nabla_x h(x, y)$ of the measurement signal. The gradient ascent leads m_t to a (often local) maximum of the signal field, which is appropriate in view of assumption (3.20). In detail, the desired dynamics for the centroid are:

$$m_{t+1} = m_t + \gamma_t g(m_t, y). \tag{3.21}$$

The main challenge is that the sensors do not have access to $g(\cdot, y)$ and can only measure a noisy version of $h(\cdot, y)$ at their current positions. Supposing noise-free measurements

[9] This assumption is made only to simplify the presentation of the gradient ascent approach. The approach generalizes to signals of higher dimension.

for now, the sensors can approximate the signal gradient at the formation centroid via a finite-difference (FD) scheme:

$$g(m_t, y) = \nabla_x h(m_t, y) = W(x_t) \begin{pmatrix} h(x_{1,t}, y) \\ \vdots \\ h(x_{n,t}, y) \end{pmatrix} - b_t, \tag{3.22}$$

where $W(x_t) \in \mathbb{R}^{d_x \times n}$ is a matrix of FD weights, which depends on the sensor states x_t, and $b_t \in \mathbb{R}^{d_x}$ captures the error in the approximation. The most natural way to obtain the FD weights is to require that the approximation is exact for a set of test functions ψ_i, $i = 1, \ldots, n$, commonly polynomials, which can represent the shape of $g(\cdot, y)$. In particular, the following relation needs to hold:

$$\begin{bmatrix} \psi_1(x_{1,t}) & \cdots & \psi_1(x_{n,t}) \\ \vdots & & \vdots \\ \psi_n(x_{1,t}) & \cdots & \psi_n(x_{n,t}) \end{bmatrix} W(x_t)^T = \begin{bmatrix} \frac{\partial}{\partial x}\psi_1(m_t) \\ \vdots \\ \frac{\partial}{\partial x}\psi_n(m_t) \end{bmatrix}, \tag{3.23}$$

where $\frac{\partial}{\partial x}\psi_i(x)$ is a row vector of partial derivatives. When $x_{i,t} \in \mathbb{R}$ the most common set of test functions are the monomials $\psi_i(x) = x^{i-1}$, in which case (3.23) becomes a Vandermonde system. The standard (monomial) FD approach is problematic when the states $x_{i,t}$ are high-dimensional and not in a lattice configuration because the system in (3.23) becomes ill-conditioned. These difficulties are alleviated by using radial basis functions (RBFs) $\psi_i(x) := \phi(\|x - x_{i,t}\|)$ as test functions. In particular, using Gaussian RBFs, $\phi(d) := e^{-(\delta d)^2}$, with *shape parameter* $\delta > 0$, guarantees that (3.23) is non-singular (Fornberg et al. 2013). Then, the FD weights obtained from (3.23) as a function of x_t are:

$$W(x_t) = R(x_t)^T \Phi(x_t)^{-T}, \tag{3.24}$$

where, for $x \in \mathcal{X}^n$, we let $\Phi_{ij}(x) := e^{-\delta^2 \|x_j - x_i\|_2^2}$ and

$$R(x) := \begin{bmatrix} 2\delta^2 e^{-\delta^2 \|x_1 - \sum_{i=1}^n x_i/n\|_2^2} (x_1 - \sum_{i=1}^n x_i/n)^T \\ \vdots \\ 2\delta^2 e^{-\delta^2 \|x_n - \sum_{i=1}^n x_i/n\|_2^2} (x_n - \sum_{i=1}^n x_i/n)^T \end{bmatrix}. \tag{3.25}$$

Since the measurements are noisy, sensor i can observe only $z_{i,t}$ rather than $h(x_{i,t}, y)$. As a result, the gradient ascent (3.21) can be implemented only approximately via $g(m_t, y) \approx W(x_t) z_t$ instead of (3.22) and with the additional complication that the measurement noise makes the iterates m_t random. Our stochastic model-free source seeking algorithm becomes:

$$m_{t+1} = m_t + \gamma_t W(x_t) z_t. \tag{3.26}$$

The convergence of similar source seeking schemes is often studied in a deterministic framework (Ögren et al. 2004) by assuming that the noise can be neglected, which is difficult to justify. In the following section, we show that the center of mass m_t, following the dynamics (3.26) with appropriately chosen step-sizes γ_t, converges to a neighborhood of a local maximum of $h(\cdot, y)$. Assuming all-to-all communication or a centralized location, which receives all state and measurement information from the sensors, the stochastic gradient ascent (3.26) can be implemented as is. It requires that the sensors are localized relative to one another, i.e., in the inertial frame of one sensor, but not globally, in the world frame. Notably, it is also not important to maintain a rigid sensor formation because the correct FD weights, necessary to combine the observations, are recomputed at every measurement location. We show that the only requirement is that the sensor team is not contained in a subspace of $\mathbb{R}^{d_x}$ when measuring (e.g., at least 3 non-collinear sensors are needed for $d_x = 2$).

Convergence Analysis

We apply the theory of stochastic approximations (Kushner and Yin 2003, Borkar 2008) to carry out the convergence analysis of the stochastic gradient ascent in (3.26). It is sufficient to consider the following stochastic approximation (SA) algorithm:

$$m_{t+1} = m_t + \gamma_t (g(m_t) + b_t + D_t), \tag{3.27}$$

where b_t is a bias term, D_t is a random zero-mean perturbation, γ_t is a small step-size, and m_t is a random sequence whose asymptotic behavior is of interest. The main result is that the iterates m_t in (3.27) asymptotically follow the integral curves of the ordinary differential equation (ODE) $\dot{m} = g(m)$. Since in our case with a fixed source state y, $g(m) := \nabla_x h(m, y)$, the ODE method of Ljung (1977) (see also Borkar 2008, Ch.2) shows that the iterates $\{m_t\}$ almost surely (a.s.) converge to the set $\{x \mid \nabla_x h(x, y) = 0\}$ of critical points of $h(\cdot, y)$ under the following assumptions[10]:

(A1) The function g is Lipschitz continuous[11].

(A2) Step-sizes $\{\gamma_t\}$ are positive scalars satisfying: $\sum_{t=0}^{\infty} \gamma_t = \infty$ and $\sum_{t=0}^{\infty} \gamma_t^2 < \infty$.

(A3) $\{D_t\}$ is martingale difference sequence with respect to the family of σ-algebras $\mathcal{F}_t := \sigma(m_0, D_s, 0 \leq s \leq t)$, i.e., D_t is measurable with respect to $\mathcal{F}_t$, $\mathbb{E}[\|D_t\|] < \infty$, and $\mathbb{E}[D_{t+1} \mid \mathcal{F}_t] = 0$ a.s. for all $t \geq 0$. Also, D_t is square-integrable with $\mathbb{E}[\|D_{t+1}\|^2 \mid \mathcal{F}_t] \leq K(1 + \|m_t\|^2)$ a.s. for $t \geq 0$ and some constant $K > 0$.

(A4) $\{m_t\}$ is bounded, i.e., $\sup_t \|m_t\| < \infty$ a.s.

(A5) $\{b_t\}$ is bounded and $b_t \to 0$ a.s. as $t \to \infty$.

The proposed source-seeking algorithm (3.26) can be converted to the SA form (3.27) as follows:

$$\begin{aligned} m_{t+1} = m_t + \gamma_t W(x_t) z_t &= m_t + \gamma_t W(x_t) \begin{pmatrix} h(x_{1,t}, y) + v_{1,t} \\ \vdots \\ h(x_{n,t}, y) + v_{n,t} \end{pmatrix} \\ &= m_t + \gamma_t \left(g(m_t, y) + b_t + W(x_t) v_t \right), \end{aligned}$$

[10] While assumptions (A1)-(A5) are sufficient to prove the convergence in our application, they are by no means the weakest possible. If necessary some can be relaxed using the results in stochastic approximation (Kushner and Yin 2003, Borkar 2008).

[11] Given two metric spaces $(\mathcal{X}, d_x)$ and $(\mathcal{G}, d_g)$, a function $g : \mathcal{X} \to \mathcal{G}$ is *Lipschitz continuous* if there exists a real constant $0 \leq L < \infty$ such that: $d_g(g(x_1), g(x_2)) \leq L d_x(x_1, x_2)$, $\forall x_1, x_2 \in \mathcal{X}$.

where the second equality follows from (3.22). Assumption (A1) ensures that $\dot{m} = g(m, y)$ has a unique solution for any initial condition and any fixed source state y. Assumption (A2) can be satisfied by an appropriate choice of the step-size, e.g., $\gamma_t = 1/(t+1)$. The selection of proper step-sizes is an important practical issue that will be discussed in the next section. We can satisfy (A4) by requiring that the environment $\mathcal{X}$ of the sensors is bounded and if necessary use a projected version of the gradient ascent (Borkar 2008, Ch.5.4). This also ensures that the FD weights are bounded and in turn (A3) is satisfied:

$$\begin{aligned}(\mathbb{E}\|D_t\|_2)^2 \leq& \mathbb{E}\|D_t\|_2^2 = \mathbb{E}\left[\|D_t\|_2^2 \mid \mathcal{F}_{t-1}\right] = \mathbb{E}\left[\|W(x_t)v_t\|_2^2\right] \\ &\leq \|W(x_t)\|_2^2 \mathbb{E}\|v_t\|_2^2 = \|W(x_t)\|_2^2 \sum_{i=1}^{n} \operatorname{tr}(\mathbb{E}[v_{i,t}v_{i,t}^T]) < \infty \\ \mathbb{E}\left[D_t \mid \mathcal{F}_{t-1}\right] =& \mathbb{E}[W(x_t)v_t] = W(x_t)\mathbb{E}v_t = 0,\end{aligned}$$

since the measurement noise in (3.19) is uncorrelated in time and has zero mean and a finite second moment. Note that the error term in (3.22) violates (A5) because it does not converge to 0. However, if we ensure that the sensor formation is not contained in a subspace of $\mathbb{R}^{d_x}$, then b_t remains bounded by some $\epsilon_0 > 0$, i.e., $\sup_t \|b_t\| \leq \epsilon_0$. Then, the argument in Borkar (2008, Ch.5, Thm.6) shows that the iterates m_t converge a.s. to a small neighborhood of a local maximum, whose size depends on ϵ_0. The result is summarized below.

Theorem 3.5. *Suppose that the gradient $g(x, y) = \nabla_x h(x, y)$ of the measurement signal is Lipschitz continuous in x, the step-sizes γ_t in* (3.26) *satisfy (A2), the sensor state space $\mathcal{X}$ is bounded and convex, and the sensor formation is not contained in a subspace of $\mathbb{R}^{d_x}$ at the measurement locations. Then, algorithm* (3.26) *converges to a small neighborhood around a local maximum of the signal field $h(\cdot, y)$.*

3.2.2 Single-robot Source Seeking

Instead of a sensor network, a *single* robot may travel to several sensing locations in order to collect the measurements needed for a finite-difference gradient approximation. Thm. 3.5 requires visiting $d_x + 1$ locations with a single robot which may result in costly maneuvers. In this section, we emphasize that the robot can use a random direction stochastic approximation (RDSA, Kushner and Yin 2003) to the gradient instead. The idea of RDSA is to estimate the gradient along a single direction at time t instead of in all d_x dimensions. We show that if the directions are chosen consistently over time, this random direction gradient ascent will still converge. In detail, we use the following gradient estimate:

$$\hat{g}_t(x_t) := \sqrt{d_x}\left[\frac{z_{t_+} - z_{t_-}}{\|x_{t_+} - x_{t_-}\|_2}\right] T_t a_t, \tag{3.28}$$

where $a_t \in \mathbb{R}^{d_x}$ is a random *direction vector* with probability distribution $p_t : \mathbb{R}^{d_x} \to [0, 1]$, and T_t is a rotation matrix, whose role is explained below. Also, $x_{t_+} := x_t + c_{t_+} T_t a_t$ and $x_{t_-} := x_t - c_{t_-} T_t a_t$ are two measurement locations determined by gain coefficients $c_{t_-}, c_{t_+} \in \mathbb{R}_{\geq 0}$. At iteration t of the algorithm, the signal field is sampled at points x_{t_-} and x_{t_+} in order to estimate its directional gradient at x_t. Then, a step is taken in the direction of the gradient estimate to generate the next estimate of the source location x_{t+1}. Repeating this procedure generates a sequence of states $\{(x_t, x_{t_-}, x_{t_+})\}$, which represents a trajectory in the world coordinate frame. The role of the matrices T_t is to transform this trajectory to the body coordinate frame of the robot. The sampling points at time t represented in the robot body frame are:

$$\begin{aligned} x^b_{t_-} &= -c_{t_-} R(-\theta_t) T_t a_t \\ x^b_{t_+} &= c_{t_+} R(-\theta_t) T_t a_t \\ x^b_{t+1} &= -\gamma_t R(-\theta_t) \hat{g}_t(0), \end{aligned}$$

where θ_t is the robot orientation and $R(-\theta_t)$ is a rotation matrix. Choosing $T_t = R(\theta_t)$, allows the robot to run the algorithm *without* the need for localization:

$$x^b_{t+1} = -\gamma_t \sqrt{d_x} \left[\frac{z_{t_+} - z_{t_-}}{\|x^b_{t_+} - x^b_{t_-}\|} \right] a_t \tag{3.29}$$

The *design parameters* of the algorithm are γ_t, c_{t_-}, c_{t_+}, and the pdf p_t of the random directions a_t. Successful application of this algorithm in a real environment requires a careful choice of the design parameters that takes the robot capabilities and the geometric constraints of the environment into account in order to ensure that the resulting sequence $\{(x_t, x_{t_-}, x_{t_+})\}$ can be followed by the robot.

Proposition 3.6. *If the sensor state space $\mathcal{X}$ is bounded and convex, the random direction stochastic gradient ascent in* (3.29) *converges a.s. to a local maximum of the signal field $h(\cdot, y)$ under the following assumptions:*

(Signal field) *$h(x, y)$ is bounded and three times continuously differentiable in x*

(Sample point gains) $c_t := \max\{c_{t_-}, c_{t_+}\} > 0,\ \ c_t \to 0$

(Gradient gains) *$\gamma_t > 0$, $\gamma_t \to 0$, $\sum_{t=0}^{\infty} \gamma_t = \infty$, and $\sum_{t=0}^{\infty} \gamma_t^2 / c_t^2 < \infty$*

(Direction vectors) *a_t are i.i.d. with distribution p_t, which is symmetric with respect to reflections about the coordinate axes, and satisfy $\mathbb{E}[\|a_t\|_2] = \sqrt{d_x}$ and $\mathbb{E}[a_t a_t^T] = I$.*

Proof. The result follows from a slight modification of the proof of the RDSA algorithm (Kushner and Yin 2003, Chapter 5.6 and 10.7) which takes into account that $c_{t_-} T_t a_t \neq c_{t_+} T_t a_t$. Note that:

$$\mathbb{E}\left[\frac{\|x_{t_+} - x_{t_-}\|_2}{\sqrt{d_x}} \right] = \frac{(c_{t_+} + c_{t_-}) \|T_t\|_2 \mathbb{E}[\|a_t\|_2]}{\sqrt{d_x}} = (c_{t_+} + c_{t_-}).$$

The rest of the assumptions are satisfied because the measurement noise v_t is zero mean and square integrable and the boundedness of $\mathcal{X}$ guarantees that $\sup_t \|x_t\| < \infty$. □

Remark. The smoothness condition on h guarantees that $\hat{g}_t$ is almost surely an unbiased estimate of g_t to within an $O(c_t^2)$ error, which is guaranteed to be small by the assumption on the sample point gains.

Since our goal is to apply the algorithm in an environment with obstacles, we disregard the assumption that the state space $\mathcal{X}$ is convex. Of course, the theoretical almost sure convergence guarantee in Prop. 3.6 is lost but we would like to show that the algorithm still works in practice with the appropriate choice of parameters. The rest of this section concentrates on the choice of γ_t, p_t, c_{t_-}, and c_{t_+} with two goals in mind: first, to take the geometric constraints of the environment and the robot characteristics into account so that the generated sampling points $\{(x_t, x_{t_-}, x_{t_+})\}$ are easy to follow and, second, to provide only a few high-level and intuitive parameters to the user of the algorithm. The parameter choice is simplified to the following two constants:

- *Aggressiveness factor* $\mathsf{r} > 0$: a constant determining the size of the steps of the algorithm. Intuitively, r is the amount of meters, by which the position of the robot is expected to change in the early iterations. The further away the source is expected to be, the larger the value of r should be.

- *Stability factor* $\mathsf{s} \geq 0$: a constant which allows for large steps in the early iterations of the algorithm without causing instabilities. It should be set to $5 - 10$ percent of the expected number of iterations of the algorithm.

Choosing the direction vectors a_t

Several choices for the pdf of a_t have been considered in literature (Spall 2003, Kushner and Yin 2003, Le Ny and Pappas 2010) with a Bernoulli distribution in each coordinate being preferred in applications. We note that the Bernoulli distribution is an optimal choice only for signal fields, which are aligned with the coordinate axes in a way that their third cross-derivatives $\partial^3 h/\partial x_i \partial x_j \partial x_k$ are all zero (Theiler and Alper 2006). Since the signal field will not be axis aligned in a non-convex $\mathcal{X}$ it is beneficial to choose a_t uniformly from all possible directions. In particular, we let p_t be a *shell distribution*, which is defined as follows: choose $a_i \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, d_x$ and then rescale the vector to guarantee that its magnitude is $\sqrt{d_x}$ as required in Prop. 3.6.

Choosing the gradient gains γ_t

The usual form used for the gradient gain coefficients in literature (Spall 2003, Kushner and Yin 2003) is:

$$\gamma_t = \frac{\gamma}{(t + 1 + \mathsf{s})^\alpha}, \; t = 0, 1, \ldots, \tag{3.30}$$

where $\gamma > 0$ is a constant, s is the stability factor mentioned earlier, and $\alpha > 0$ governs the decay rate for the gains and can be set to $\alpha = 0.602$ as suggested in (Spall 2003, Ch.6).

A modification to this choice is required for our application. If the numerator γ is constant, the gain coefficients γ_k are monotonically decreasing, which is not desirable because the robot will be taking decreasing steps along the gradient and it might get trapped in a location, where the magnitude of the gradient estimate is small. We replace γ with a time-varying numerator $\gamma_t^0 > 0$, which is inversely proportional to the magnitude of the gradient estimate. This is beneficial because when the magnitude of the gradient estimate is large, the robot takes small steps in a controlled manner towards the source but if the magnitude of the gradient estimate decreases, the gain coefficients increase allowing the robot to follow the gradient even if the signal field is very flat. Based on these observations we propose:

$$\gamma_t^0 = \frac{\mathsf{r}(1+\mathsf{s})^{\alpha}}{\displaystyle\frac{1}{w}\sum_{j=t+1-w}^{t}\frac{1}{d_x}\|\hat{g}_j(x_j)\|_1},\ t = 0, 1, \dots, \tag{3.31}$$

where r is the aggressiveness factor and $w \in \mathbb{N}$ is a window over which the mean magnitude of the elements of $\hat{g}_t$ is averaged. The size of w determines the speed at which γ_t reacts to changes in the magnitude of the gradient estimate. We used $w = 10$ in the experiments in Sec. 3.2.4.

Choosing the sample point gains c_{t_-} and c_{t_+}

A typical schedule used in the stochastic approximation literature (Spall 2003, Kushner and Yin 2003) for the sample point gain coefficients is:

$$c_{t_-} = \frac{c_-^0}{(t+1)^{\beta}}, \qquad c_{t_+} = \frac{c_+^0}{(t+1)^{\beta}},\ \ t = 0, 1, \dots, \tag{3.32}$$

where $c_-^0, c_+^0 \in \mathbb{R}_{\geq 0}$ are constants and $\beta \in \mathbb{R}_{>0}$ is the gain decay rate. In practical applications, a slow decay leads to better finite sample performance and a good choice is $\beta = 0.101$ (see Spall 2003). The constants c_-^0 and c_+^0 are typically set to the standard deviation of the measurement noise v_t at the current position of the robot by measuring the signal several times. The gains c_{t_-}, c_{t_+}, and the direction vector a_t affect the position of the sampling points x_{t_-} and x_{t_+} as specified in (3.28). While the choice in (3.32) is applicable to the obstacle-free case, when dealing with a general environment it needs to be modified to accommodate for the constraints introduced by the obstacles. Let b_{t_+} be the value for c_{t_+} originally suggested in (3.32). Suppose that the robot is traveling from its previous estimate x_{t-1} towards x_t. As soon as x_t is in the robot's field of view $\mathcal{F}$, we choose x_{t_+} in $\mathcal{F}$ to ensure that it is reachable. Alg. 8 with $x \leftarrow x_t$ and $r \leftarrow b_{t_+}$ shows how to sample a_t and simultaneously choose c_{t_+}, with a magnitude as close to b_{t_+} as possible.

Algorithm 8 Rejection Sampling of a_t

```
1: **Input**: Position $x \in \mathbb{R}^{d_x}$ in the robot body frame, radius $r \in \mathbb{R}_{\geq 0}$, and field of view $\mathcal{F} \subseteq \mathsf{X}$
2: **Output**: Step size $c_{t_+}$ and direction vector $a_t$
3: Let $dS$ be a small area element on the surface of the hypersphere of radius $\sqrt{d_x}$ centered at $x$
4: $count \leftarrow 1$
5: **repeat**
6:     Sample $a_t$ from the shell distribution
7:     **if** $(1 - \mathbb{P}(\{a_t \in dS\}))^{count} < 0.05$ **then**
8:         decrease $r$; $count \leftarrow 1$
9:     **else**
10:        $count \leftarrow count + 1$
11: **until** $r = 0$ **or** there is a path between $x$ and $(x + r a_t)$ in $\mathcal{F}$
12: **return** $a_t$ and $c_{t_+} \leftarrow r$
```

The shell distribution is sampled for a direction a_t, which selects a possible sample point $q = x_t + r a_t$. Line 11 checks if the path from x_t to q is within the robot's field of view and if so the chosen values for a_t and $c_{t_+} \leftarrow r$ are returned. Otherwise, another sample for a_t is chosen. Thus, the allowable values for $x_{t_+} = x_t + c_{t_+} a_t$ lie on the intersection of the field of view $\mathcal{F}$ and the ball of radius $r\sqrt{d_x}$ centered at x_t. Due to the obstacles, there might not be a feasible choice for a_t with the specified radius r. When the probability of selecting a sampling point in any small region dS on the surface of the sphere of radius r is above 95% but a suitable direction has not been chosen yet, the radius is decreased (line 6). Once the direction a_t is known, the choice of c_{t_-} is the maximum distance that can be traveled along $-a_t$ starting from x_t and up to b_{t_+} or until an obstacle is reached.

3.2.3 Distributed Multi-robot Source Seeking

Now, we return to the case when a sensor team is available to estimate the complete gradient in (3.26) instead of the directional one. As mentioned earlier, if the sensors are localized in the frame of reference of one of them and if all-to-all communication is available, the algorithm can be implemented as is. However, in many scenarios, all-to-all communication is either infeasible or prone to failures and sensor localization needs to be carried out online. In this section, we describe a distributed implementation of the model-free algorithm (3.26), in which the sensors use relative measurements of their neighbors' states to estimate the collective state of the formation. In detail, the sensors need to estimate the formation state x_t, the centroid m_t, and the stochastic approximation to the signal gradient $W(x_t)z_t$ at each time t using only local information. We introduce a fast time-scale $k = 0, 1, \ldots$, which will be used for the estimation procedure at each measurement location, i.e., at each time t. During the formation state and gradient estimation at time t, the sensors remain stationary and we drop the index t to simplify the notation.

Let the communication network of the n sensors be represented by an undirected graph $G = (\{1, \ldots, n\}, E)$. Suppose that each sensor i receives a relative measurement of the state of each of its neighbors $j \in \mathsf{N}_i$:

$$s_{ij}(k) = x_j - x_i + \epsilon_{ij}(k), \quad \epsilon_{ij}(k) \sim \mathcal{N}(0, E_{ij}), \tag{3.33}$$

where $\epsilon_{ij}(k)$ is the measurement noise which is independent at any pair of times on the fast time-scale and across sensor pairs. If each sensor manages to estimate the states of the whole sensor formation using the measurements $\{s_{ij}(k)\}$, then each can compute the finite-difference weights in (3.24) on its own.

The distributed linear Gaussian estimator (2.25) in Sec. 2.6.1 can be employed to estimate the sensor states x. Notice that it is sufficient to estimate x in a local frame because neither the finite difference computation (3.24) nor the gradient ascent (3.26) requires global state information. Assume that all sensors know that sensor 1 is the origin at every measurement location. Let $x^* := \begin{bmatrix} 0^T & (x_2 - x_1)^T & \cdots & (x_n - x_1)^T \end{bmatrix}^T$ denote the true sensor states in the frame of sensor 1. Let $\hat{x}^i(k)$ denote the estimate that sensor i has of x^* at time k on the fast time scale. The vector form of the measurement equations (3.33) is:

$$s(k) = (B \otimes I_{d_x})^T x^* + \epsilon(k), \tag{3.34}$$

where B is the incidence matrix of the communication graph G. The measurements (3.34) fit the linear Gaussian model in (2.22). Since the first element of x^* is always 0, only

$(n-1)d_x$ components need to be estimated. As the rank of $B \otimes I_{d_x}$ is also $(n-1)d_x$, Thm. 2.10 allows us to use the distributed estimator (2.25) to update $\hat{x}^i(k)$.

Concurrently with the state estimation, sensor i would be obtaining observations $z_{i,t}(k)$ of the signal field for $k = 0, 1, \ldots$[12]. In the centralized case (Sec. 3.2.1), each sensor uses the following gradient approximation:

$$g(m_t, y) \approx W(x_t)z_t = \sum_{i=1}^{n} \mathbf{col}_i(W(x_t))z_{i,t}, \tag{3.35}$$

where $\mathbf{col}_i(W(x_t))$ denotes the ith column of the finite-difference-weight matrix. Since x_t and z_t are not available in the distributed setting, each sensor can use its local measurements $z_{i,t}(k)$ and its estimate $\hat{x}_t^i(k)$ of the sensor states to form its own local estimate of the signal gradient:

$$\hat{g}_{i,t}(k) := \mathbf{col}_i(W(\hat{x}_t^i(k)))\frac{1}{k+1}\sum_{\tau=0}^{k} z_{i,t}(\tau). \tag{3.36}$$

In order to obtain an approximation to $g(m_t, y)$ as in (3.35) in a distributed manner, we use the high-pass dynamic consensus filter of Spanos et al. (2005) to have the sensors agree on the value of the sum:

$$\hat{g}_t(k) := n\left(\frac{1}{n}\sum_{i=1}^{n} \hat{g}_{i,t}(k)\right).$$

Each node maintains a state $q_{i,k}$, receives an input $\mu_{i,k}$, and provides an output $r_{i,k}$ with the following dynamics:

$$\begin{aligned} q_{i,k+1} &= q_{i,k} + \beta \sum_{j \in \mathsf{N}_i} (q_{j,k} - q_{i,k}) + \beta \sum_{j \in \mathsf{N}_i} (\mu_{j,k} - \mu_{i,k}) \\ r_{i,k} &= q_{i,k} + \mu_{i,k} \end{aligned} \tag{3.37}$$

where $\beta > 0$ is a step-size. For a connected network Spanos et al. 2005, Thm. 1 guarantees that $r_{i,k}$ converges to $1/n \sum_i \mu_{i,k}$ as $k \to \infty$. The following result can be shown by letting $\mu_{i,k} := \hat{g}_{i,t}(k)$ and is proved in Appendix D.15.

Theorem 3.7. *Suppose that the communication graph G is strongly connected. If the sensor nodes estimate their states x^* from the relative measurements (3.34) using algorithm* (2.25)*, compute the finite-difference weights* (3.24) *using the state estimates, and run the dynamic consensus filter* (3.37) *with input $\mu_{i,k} := \hat{g}_{i,t}(k)$, which was defined in* (3.36)*, then the output $r_{i,k}$ of the consensus filter satisfies:*

$$n\left(\lim_{k\to\infty} \mathbb{E}[r_{i,k}]\right) = g(m^*, y) + b, \quad \forall i \in \{1, \ldots, n\},$$

where $g(m^*, y)$ is the true signal gradient at $m^* := \sum_{i=1}^{n} x_i^*/n$ and b is the error in the finite-difference approximation (3.22).

[12] The time-scales of the relative state measurements and the signal measurements might be different but for simplicity we keep them the same.

After this procedure the sensors agree on a centroid for the formation and a gradient estimate, which can be used to compute the next formation centroid according to (3.26). Since the finite-difference weights are recomputed at every t, the formation need not be maintained accurately. This allows the sensors to avoid obstacles and takes care of the motion uncertainty.

3.2.4 Application: Wireless Radio Source Seeking

This section demonstrates the performance of the source-seeking algorithms in a wireless radio source localization task. A radio signal was chosen because it is very noisy and difficult to model and yet most approaches for wireless source seeking are model-based. Both simulations and real world experiments were carried out. We begin by modeling the received signal strength (RSS), which is needed to simulate the radio signal.

Wireless Radio Signal Model

Let the positions of a wireless source and receiver in 2-D be y and x, respectively. The *received signal strength* (dBm) at x is modeled as:

$$P_{rx}(x,y) = P_{tx} + G_{tx} - L_{tx} + G_{rx} - L_{rx} - L_{fs}(x,y) - L_m(x,y) - R(x,y),$$

where P_{tx} is the transmitter output power (18 dBm in our experiments), G_{tx} is the transmitter antenna gain (1.5 dBi), L_{tx} is the transmitter loss (0 dB), G_{rx} is the receiver antenna gain (1.5 dBi), L_{rx} is the receiver loss (0 dB), L_{fs} is the free space loss (dB), L_m is the multi-path loss (dB), and R is the noise. The free space loss is modeled as:

$$L_{fs}(x,y) = -27.55 + 20\log_{10}(\nu) + 20\log_{10}\big(\|x-y\|_2\big),$$

where ν is the frequency (2400 MHz). The model from Capulli et al. (2006) is used for the the multi-path loss:

$$L_m(x,y) = \begin{cases} \alpha + \beta\lambda(x,y), & \text{if } \lambda(x,y) > 0 \\ 0, & \text{else} \end{cases}$$

where α is a multi-wall constant (30 dB), β is a wall attenuation factor (15 dB/m), and $\lambda(x,y)$ denotes the distance traveled by the ray from y to x through occupied cells in the environment (represented as an occupancy grid). Finally, if the measurement is line-of-sight (LOS), i.e., $\lambda(x,y) = 0$, the fading $R(x,y)$ is Rician(μ,σ); otherwise it is Rayleigh(σ). We used $\mu = 4$ dB and $\sigma = 20$ dB in the simulations.

Robot Platform

We carried out simulations and real-world experiments in an indoor environment using a differential-drive Scarab robot (Fig. 3.22). The robot was equipped with a Hokuyo URG laser range finder for obstacle detection and a XBee-PRO RF module for RSS measurements. An A^* planner was used to generate a feasible path along the sampling points supplied by the stochastic gradient ascent algorithm. The robots were controlled by simulating possible trajectories over a short horizon and evaluating them based on orientation correctness, proximity to the planned path, speed, obstacle avoidance, and distance to the next sampling

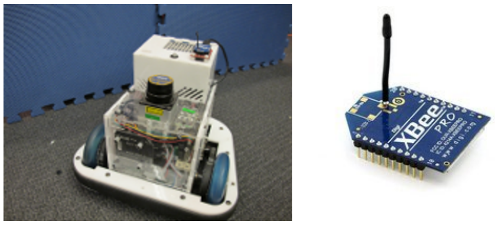


Figure 3.22: A differential-drive Scarab robot equipped with a XBee-PRO RF module.

point. Two Scarabs were used in the single-robot experiments: one transmitting a wireless signal and the other acting as the seeker. A team of 10 robots was used in the multi-robot simulations.

Single-robot Simulations

Simulations were carried out in an obstacle-free environment using the radio signal model described above. The positions of the source and the seeker as well as a sample trajectory after 20 iterations of the random-direction stochastic approximation algorithm (3.29) are shown in Fig. 3.23. The results confirm the convergence guarantees of the RDSA algorithm in a convex environment (Prop. 3.6) empirically. Starting 21.85 meters away and running the algorithm for only 30 iterations resulted in a mean error in the source estimate of less than 30 cm. For these simulations we used $\mathsf{r} = 1.5$, $\mathsf{s} = 4$, and $w = 10$.

Next, we applied the RDSA algorithm in non-convex environments, for which convergence to the source cannot be guaranteed theoretically. Fig. 3.24 shows an example trajectory followed by the source seeker. Fifty independent replications were simulated for several different run lengths to investigate the finite-sample performance. We can see that, as the number of iterations increases, the mean distance from the estimated source position to the actual one decreases. The variance in the source position estimate in this case is much larger than in the obstacle-free case because the robot can get trapped in local maxima created by the obstacles. The parameters in this case were $\mathsf{r} = 2.5$, $\mathsf{s} = 7$, and $w = 10$.

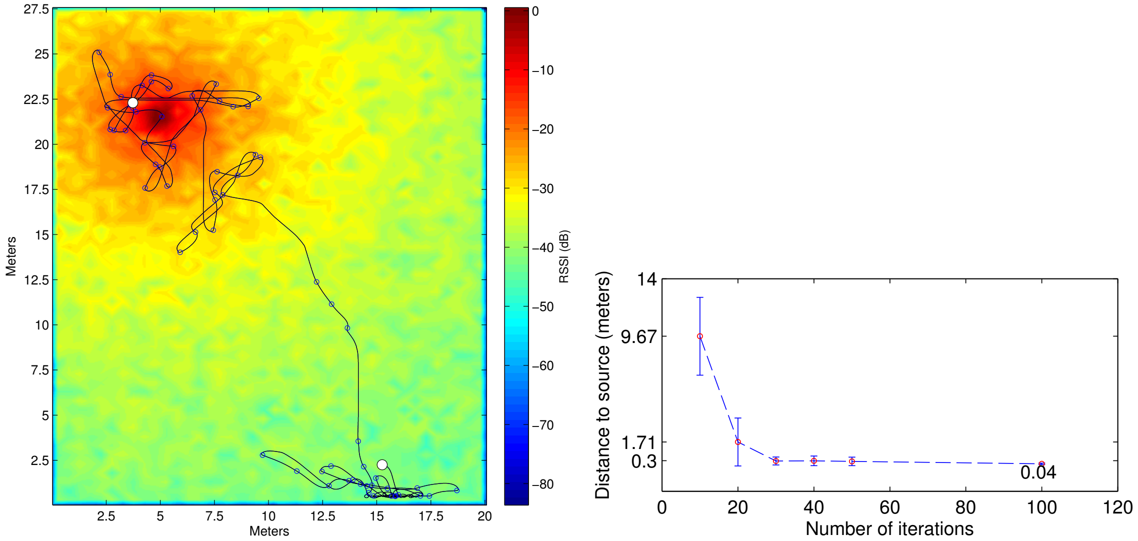


Figure 3.23: A path followed by the robot after 20 iterations of the random-direction stochastic approximation algorithm (3.29) in an obstacle-free environment is shown on the left. The blue circles indicate positions at which the robot measured the signal strength. The white dots indicate the starting and final positions of the robot and are 21.85 m and 1.47 m away, respectively, from the actual position of the source. The right plot shows the final distance to the source calculated over 50 independent replications. The source was 21.85 m away initially. The bars show one standard deviation.

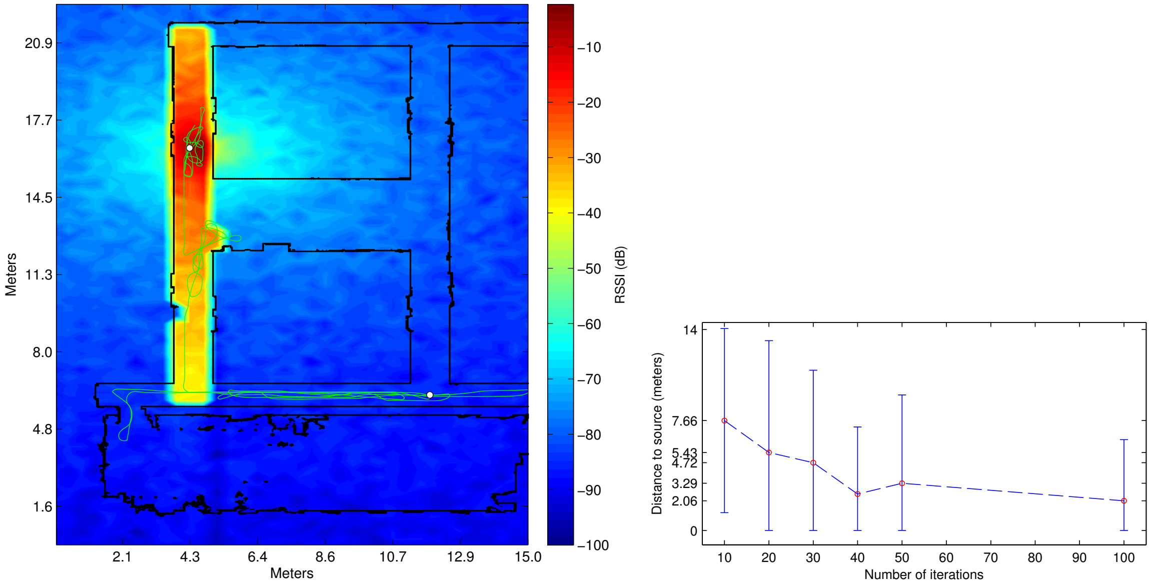


Figure 3.24: A path followed by the robot after 20 iterations of the random-direction stochastic approximation algorithm (3.29) in an environment with obstacles is shown on the left. The white dots indicate the starting and the final positions. The seeker and the source were 17.85 m apart initially and 0.75 m apart in the end. The distance to the source calculated over 50 independent replications of the algorithm is shown on the right. The initial distance from the source was 17.85 m. The bars show one standard deviation from the mean.

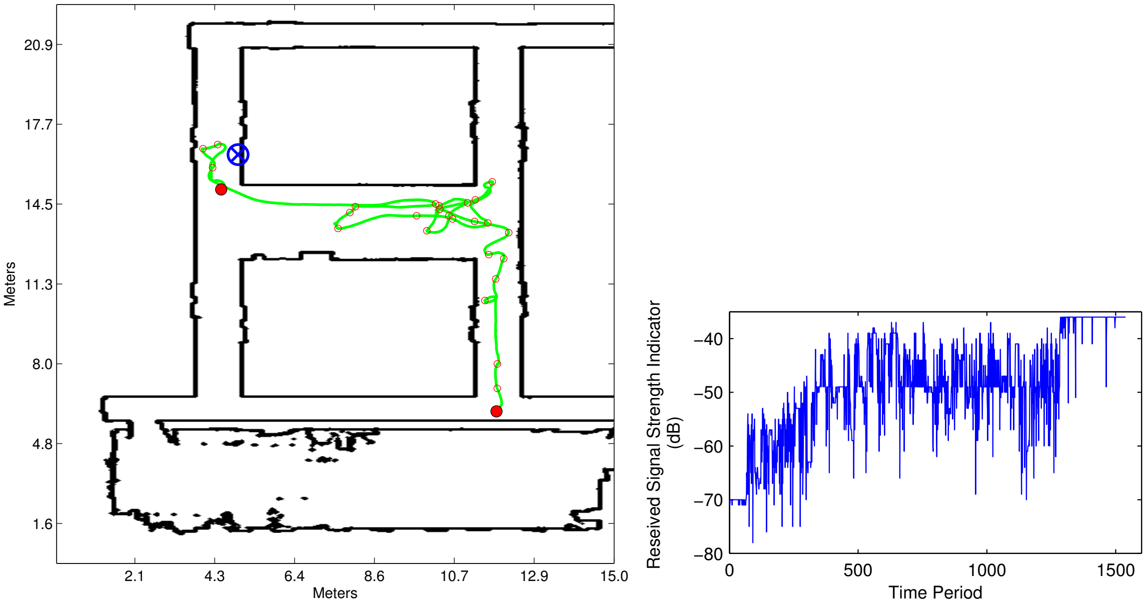


Figure 3.25: Path followed by the robot after 10 iterations of the random-direction stochastic approximation algorithm (3.29) in a real environment. The seeker was 17.9 m away from the source initially. The red circle shows the final estimate of the source location and is 2.2 m away from the actual one, denoted by the blue cross. The received-signal-strength measurement history is shown on the right. See Appendix D.16 Extension 15 for a video of the experiment.

Singe-robot Experiments

The simulations show that the finite-time performance of the RDSA algorithm (3.29) is good even in non-convex environments but suffer from one drawback. The radio signal model is not very accurate for an indoor environment because temporal and spatial fluctuations in the signal due to fading and shadowing are significant Goldsmith (2005). We carried out real-world experiments to evaluate the performance of the stochastic gradient ascent on a real wireless signal. Fig. 3.25 shows the map of the environment and the trajectory followed by the robot after 10 iterations of the random-direction stochastic approximation algorithm (3.29) in one of the experiments. The parameters were $\mathsf{r} = 3.5$, $\mathsf{s} = 4$, and $w = 10$. Several experiments with different source and seeker starting positions were performed in the same environment. While we do not have enough data to reproduce the extensive convergence analysis in the simulations, the experiments showed that the algorithm is applicable to real signals. The robot's movement is less erratic when further away from the source because the change in the gradient is significant and easy to discern. Closer to the source, however, the signal field is relatively flat and the measurement noise affects the gradient estimation significantly. See Appendix D.16 Extension 15 for a video of the experiment.

Multi-robot Simulations

The multi-robot simulations were performed in an obstacle-free environment. First, we verified the conclusions of Thm. 3.7 when the sensor formation is not maintained well, namely that the distributed relative pose estimation and the consensus on the local finite-difference gradient estimates converge asymptotically to an unbiased (up to the error in

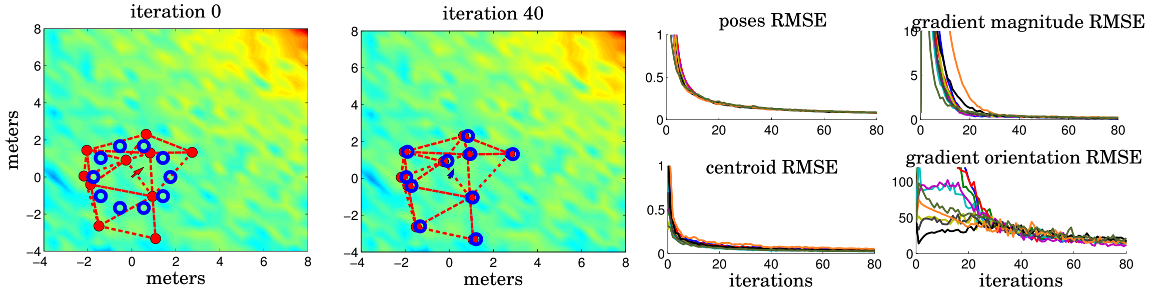


Figure 3.26: Joint position and gradient estimation at a single measurement location (on the fast time-scale). The first plot shows the true sensor positions (red circles), initial position estimates (blue circles), and the true gradient of the signal field (red arrow). The second plot shows the position estimates after 40 iterations (blue circles) and the gradient estimate of sensor 1 (blue arrow). The third column shows the root mean squared error (RMSE) of the position (top) and centroid (bottom) estimates of all sensors averaged over 50 independent repetitions. The forth column shows the RMSE of the gradient magnitude and orientation estimates.

the finite-difference approximation) gradient estimate. Ten sensors were arranged in a distorted "circular" formation (see Fig. 3.26) and were held stationary during the estimation procedure on the fast time-scale. Initially, the sensors assumed that they were in a perfect circular formation of radius 1.75 meters. Relative measurements (3.33) with noise covariance $E_{ij} = 0.4I_2$ were exchanged to estimate the sensor states. At each time k, sensor i used its estimate $\hat{x}^i(k)$ to compute the finite-difference weights via (3.24). Wireless signal measurements, obtained according to the RSS model, were combined with the finite-difference weights to form the local gradient estimates (3.36), which were used to update the state of the consensus filter according to (3.37). Fig. 3.26 shows that the errors in the pose and the gradient estimates tend to zero after 80 iterations on the fast time-scale.

Next, we demonstrate the ability of the stochastic gradient ascent algorithm to localize the source of a wireless radio signal obtained according to the received-signal-strength model. The performance of the model-free algorithm is illustrated in Fig. 3.27. A circular formation with radius 1.75 meters consisting of 10 sensors was maintained. The robots had a communication radius of 6 meters. They did not coordinate to maintain the formation but were kept together by the agreement on the centroid and the signal gradient, achieved via the distributed state estimation and the consensus filter. At time t, each sensor i applied the control $u_{i,t} = \gamma_t \hat{g}_{i,t}(K_{max})$, where $\hat{g}_{i,t}(K_{max})$ is the gradient estimate after $K_{max} = 50$ iterations on the fast time-scale and γ_t is the step-size. Unlike the persistent measurements in Fig. 3.26, the sensors measured their relative states and the wireless signal only 10 times and stopped updating their local gradient estimates to enable faster convergence of the consensus filter. The initial distance between the source and the centroid of the sensor formation was 44.2 meters. Averaged over 50 independent repetitions the sensors managed to estimate the source location to within 4.62 meters in 30 iterations.

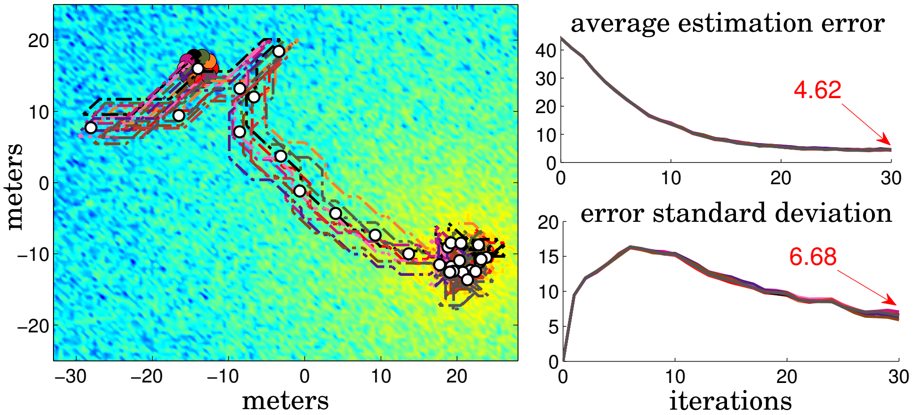


Figure 3.27: The paths followed by the sensors after 30 iterations of the model-free source-seeking algorithm in an obstacle-free environment. The white circles indicate sensor 1's estimates of the source position over time. The plots on the right show the average error of the source position estimates and its standard deviation averaged over 50 independent repetitions.

3.3 Summary

This chapter considered two specific cases of practical importance in which the linear Gaussian techniques of Ch. 2 cannot be applied. The first was a semantic localization problem whose difficulty was due to discrete (object class) measurements, false and missed detections, and unknown data association. Modeling the semantic information obtained from object recognition with random finite sets enabled a unified treatment of these sensing nuisances. It was shown that the resulting semantic observation model can be evaluated via a matrix permanent computation. This critical connection with the permanent enabled the efficient use of the observation model for Bayesian filtering. Applications in vehicle localization in residential areas, global localization of Google's Tango phone, and robot localization from detected objects demonstrated precise and robust localization from semantic information in various scenarios and over many repetitions. Compared to maximum likelihood data association, out approach was more robust to perceptual aliasing and offered superior performance in cases of global localization and loop-closure. Finally, to enable autonomous localization, we addressed the active semantic localization problem via greedy minimization of the observer's pose entropy, conditioned on future semantic measurements. Evaluating the entropy criterion turned out to be particularly challenging due to the non-Gaussian distributions and the unknown data association. The simulations demonstrated that this approach, although computationally demanding, outperforms simpler active localization heuristics.

In the second case, we considered localizing the source of a physical signal of interest using a team of mobile sensors. The main challenge was that the sensor observation models were completely unknown. We proposed a stochastic finite-difference approximation to the signal gradient and proved that following the gradient in a convex environment would lead the sensor team to a local maximum of the sensed signal. Our model-free algorithm does not require global localization of the sensors and is robust to deformations in the geometry of the team. Further, we discussed a distributed version of the algorithm, in which the sensors estimate their poses using relative measurements from their neighbors and reach a consensus on the gradient estimate. To avoid costly maneuvers when only a single robot is available, we proposed estimating a *directional* gradient of the signal, which requires visiting just two measurement locations. We discussed parameter choices which enable good practical performance even in environments with obstacles. The stochastic gradient ascent algorithms were demonstrated in several wireless radio source seeking experiments.

In sum, this chapter discussed situations in which the target inference could not be performed using standard estimation techniques, which forced us to apply greedy information-seeking control. Nonmyopic closed-loop planning in a non-Gaussian setting will be discussed in the next chapter.

Chapter 4

Nonmyopic Information Acquisition with General Discrete Models

While external techniques can be used to handle missed detections, false alarms, and unknown data association, when the target state or measurement spaces are discrete, the models cannot be linearized and the linear-Gaussian technique for informative planning described in Ch. 2 is not applicable. As we saw in Ch. 3, such complications arise in object recognition applications. Unfortunately, removing the linear-Gaussian assumptions invalidates the separation principle (Thm. 2.1) and a closed-loop policy is needed to solve the original stochastic active information acquisition problem (1.4). Unlike Ch. 3 which uses greedy planning, this chapter focuses on nonmyopic closed-loop planning for information gathering in discrete state and measurement spaces.

An algorithm, which computes the optimal control policy, can be obtained by considering the discrete-space active information acquisition problem as an instance of a partially-observable Markov decision process (POMDP) and applying backward value iteration (Bertsekas 1995). Albeit optimal, backward value iteration, scales poorly with the cardinality of the state space. Instead of the exact algorithm, in Sec. 4.4 we use an offline point-based approximate POMDP solver (Kurniawati et al. 2008) to obtain a non-greedy camera control policy in the context of active object classification and pose estimation. Still, a general POMDP solver does not exploit the properties of the particular cost function (i.e., the information measure) and, being an offline approach, has difficulties handling various constraints (e.g., revisiting viewpoints or avoiding occlusions) and changing environments. For these reasons, in Sec. 4.3 we propose an *online approximate* algorithm for nonmyopic closed-loop informative planning based on Monte Carlo tree search (Browne et al. 2012), which can handle various constraints and has exceptional performance in large challenging domains such as game solving (Gelly and Silver 2007) and belief-space planning in robotics (Hauser 2011, Nguyen et al. 2014, Lauri and Ritala 2014). We use the tree search algorithm in conjunction with a rollout policy[1], specifically designed to exploit the properties of the information (or rather uncertainty) measure.

This chapter is based on the papers Atanasov et al. (2013, 2014b), Lauri et al. (2015), Zhu et al. (2014) and presents applications in camera view planning for object classification

[1] The rollout policy in Monte Carlo tree search plays a similar role as the base policy in policy iteration (Bertsekas 1995)

and pose estimation (Sec. 4.4) and in single-image object recognition via deformable part models (Sec. 4.5). In the language of active information acquisition, the class and pose of an unknown object constitute the target state, while the probability of having misclassified objects and incorrect orientation estimates is used as the uncertainty measure. Rather than placing the burden of providing perfect recognition results on a static sensor, in Sec. 4.4 we consider controlling the viewpoint of a mobile camera in order to gain more information about the scene and thus increase the confidence in the object recognition results. This can also mitigate the effects of occlusions, lighting variations, and imperfect object models. Our optimization criterion aims to balance the detection accuracy with the time spent moving the sensor to different viewpoints. This encodes the requirements of an object recognition task more precisely than a criterion based on mutual information.

Some of the earliest work in active perception is by Bajcsy (1988), Krotkov and Bajcsy (1993). It is focused on improving the estimates of objects' 3D positions by controlling the intrinsic parameters of a stereo camera. Similarly, Pito (1999) addresses the problem of selecting the resolution of a camera to improve surface reconstruction by maximizing information gain. Since then, many active vision works have utilized information-theoretic criteria for viewpoint and sensor parameter selection. Sommerlade and Reid (2008) control a pan-zoom-tilt camera with a fixed position to track mobile targets based on myopic minimization of conditional entropy. This simplifies the problem considerably because the trade-off between minimizing the sensor movement energy and maximizing viewpoint informativeness is avoided. Methods, which deal with active selection of views for realistic sensor models typically resort to myopic planning (Browatzki et al. 2012, Borotschnig et al. 2000, Potthast and Sukhatme 2014). Denzler and Brown (2002) select the focal length, pan and tilt angles, and the viewpoint of a camera on a hemisphere around an object of interest via greedy maximization of entropy. Borotschnig et al. (2000) represent object appearance via parametric eigenspaces and use probability distributions in the eigenspace to greedily select discriminative views. Eidenberger and Scharinger (2010) use a mobile camera to classify stationary objects and estimate their poses. Static detection is performed using SIFT matching and the object pose distributions are represented with Gaussian mixtures. Similar to our approach, the problem is encoded by a POMDP but instead of an approximate nonmyopic policy, the authors resort to a greedy approach to reduce the differential entropy in the object pose and class distributions. Karasev et al. (2012) plan the path of a mobile sensor for visual search of an object in an otherwise known and static scene. The authors hypothesize about the object's pose and apply greedy maximization of the conditional entropy of the next measurement. Paletta and Pinz (2000) describe a reinforcement learning approach for obtaining a sequence of views which maximally discriminate objects of various classes at different orientations. An approximate policy which maps a sequence of received measurements to a discriminative view point is computed offline. Velez et al. (2012) consider detecting doorways along the path of a mobile sensor traveling towards a fixed goal. The unknown state of a candidate detection is binary: "present" or "not present". Deformable part models (Felzenszwalb et al. 2010b) are used for static detection; stereo disparity and plane fitting for pose estimation. An entropy field is computed empirically offline for all viewpoints in the workspace and is used to nonmyopically select locations with high information gain. The planning is open-loop, however, because the object state distributions change online but only the precomputed entropy field is used.

Inspired by Velez et al. (2012) and the work on hypothesis testing (Naghshvar and Javidi 2013b, Sankaran 2012), we introduce a rough discretization of the space of orientations so that the hidden object state takes on several values: one for "object not present" and the rest for "object present" with a specific orientation. Another approach based on hypothesis testing is Laporte and Arbel (2006). To disambiguate competing hypotheses the authors myopically select views to maximize the dissimilarity between the distributions of the expected measurements. In recent years, the active vision problem has received significant attention in the robotics community as well. Hanheide et al. (2011) present an approach for object search and place categorization in large indoor environments. A probabilistic model is used to encode structural relations among objects and places (e.g., cereal boxes are often located in kitchens). An object search task is then represented by pairing the probabilistic conceptual map with the visual appearance of an object of interest. A sequence of views is planned using a POMDP abstraction with conditional entropy as the cost function (Göbelbecker et al. 2011). Other recent work which does active visual search in a similar spirit is Aydemir et al. (2011). Probabilistic spatial relations and static properties of rooms are used to pose the problem as a fully-observable Markov decision process (MDP). A greedy next-best-view approach is used to determine if an object is present or not at a specific location, while the MDP is used to synthesize a sequence of good locations to search. Sridharan et al. (2010) plan visual sensing actions for scene understanding and disambiguation. Similar to our approach, a POMDP captures the trade-off between plan reliability and execution time and enables a robot to simultaneously decide which region in the scene to focus on and what processing to perform.

Among the contributions of this chapter is a new 3D object detector, the viewpoint-pose tree (VP-Tree, Sec. 4.4.1), that uses pointcloud data from a depth sensor to provide a *pose estimate* in addition to detecting an object's class. This is achieved via partial view matching and helps in cases when the object is partially occluded or in contact with another object. Real-world experiments were carried out to recognize objects of interest by controlling the trajectory of a RGB-D camera attached to the arm of a PR2 robot. The results show that nonmyopic closed-loop planning outperforms the traditional greedy next-best-view approach.

Instead of controlling the camera viewpoint, Sec. 4.5 uses active information acquisition to speed-up object recognition in a *single* image via part-based models. Part-based models, such as deformable part models (DPM, Felzenszwalb et al. (2010b)), represent an object as a collection of parts (e.g., a car is represented by a windshield, door, wheels, etc.) and corresponding filters, based on local features. DPM-based detectors achieve unrivaled accuracy on standard datasets but their computational demand is high since it is proportional to the number of parts in the model and the number of (pixel-scale) image locations at which the part filters are evaluated. Approaches for speeding-up DPM inference such as cascades, branch-and-bound, and multi-resolution schemes, use the response obtained from the initial filter evaluation to reduce future computations. While these approaches rely on a predetermined sequence of parts, we can use active information acquisition to optimize the order in which to apply the part filters and the time at which to stop and predict an object label. In the language of information acquisition, the sensor state consists of the set of previously-used parts, the target state is whether an object is present or not at a specific image location, the observations consist of the part filter scores, and the uncertainty measure captures the probability of choosing an incorrect target label. Since the target and measurement spaces are sufficiently small, the optimal nonmyopic closed-loop policy for ordering the part filters and choosing the stopping time can be computed via backward value iteration. Our active DPM approach demonstrates a speed-up of 3 times compared to cascade DPM (Felzenszwalb et al. 2010a) with negligible loss in accuracy when evaluated on the PASCAL VOC 2007 and 2010 datasets (Everingham et al. 2010). The approach is independent of the representation and can generalize to any classification problem that involves linear additive scores and uses several evaluation stages (e.g., parts filters).

The closest approaches in the literature include Felzenszwalb et al. (2010a), Sznitman et al. (2013), Wu and Zhu (2013), Gao and Koller (2011), Karayev et al. (2014). Sznitman et al. (2013) maintain a foreground probability at each stage of a multi-stage ensemble classifier and determine a stopping time based on the corresponding entropy. Wu and Zhu (2013) learn a sequence of score thresholds for stopping by minimizing an empirical loss function. Gao and Koller (2011) optimize the order of applying ensemble classifiers by greedily choosing the next classifier that minimizes entropy. Karayev et al. (2014) propose anytime recognition via Q-learning given a computational cost budget. In contrast to existing work, our approach optimizes the stage order and the stopping criterion jointly. Kokkinos (2011) used branch-and-bound to prioritize the search over image locations driven by an upper bound on the classification score. It is related to our approach in that object-less locations are easily detected and the search is guided in (image) location space but with the difference that our policy proposes the next part to be tested in cases when no label can yet be given to a particular location. Earlier approaches (Lampert et al. 2008, Lehmann et al. 2011b, Lampert 2010) relied on branch-and-bound to constrain the search space of object detectors based on a sliding window or a Hough transform but without deformable parts. Another related group of approaches focuses on learning a sequence of object template tests in position, scale, and orientation space that minimizes the total computation time through a coarse-to-fine evaluation (Fleuret and Geman 2001, Pedersoli et al. 2011).

The classic work of Viola and Jones (2001) introduced a cascade of classifiers whose order was determined by importance weights, learned by AdaBoost. The approach was studied extensively in Bourdev and Brandt (2005), Brubaker et al. (2008), Gualdi et al. (2012), Lehmann et al. (2011a), Zhang et al. (2011). Recently, Dollár et al. (2012) introduced cross-talk cascades which allow detector responses to trigger or suppress the evaluation of weak classifiers in the neighboring image locations. Weiss et al. (2012) used structured prediction cascades to optimize a function with two objectives: pose refinement and filter evaluation cost. Sapp et al. (2010) learn a cascade of pictorial structures with increasing pose resolution by progressively filtering the pose-state space. Its emphasis is on pre-filtering structures rather than part locations through max-margin scoring, so that human poses with weak individual part appearances can still be recovered. Rahtu et al. (2011) used general "objectness" filters in a cascade to maximize the quality of the locations that advance to the next stage. Our approach is also related to and can be combined with active learning via Gaussian processes for classification (Kapoor et al. 2010).

4.1 Active Information Acquisition in Discrete Spaces

As discussed earlier, one important situation in which the linearization technique of Ch. 2 cannot be applied is when some measurement or state variables are discrete. Therefore, in this chapter we add the following assumption to the general active information acquisition problem (1.4).

Assumption 4.1. *The sensor state space $\mathcal{X}$, the target state space $\mathcal{Y}$, and the measurement space $\mathcal{Z}$ are finite sets.*

Apart from this restriction, we proceed with very general motion and observation models. As before, let the sensor state at time t be $x_t \in \mathcal{X}$ and let the sensor motion model be:

$$x_{t+1} = f(x_t, u_t), \quad u_t \in \mathcal{U}$$

where $\mathcal{U}$ is a finite space of admissible control inputs. Let the sensor observation model and the target motion model be described by general probability mass functions (pmfs) over $\mathcal{Z}$ and $\mathcal{Y}$:

$$\begin{aligned} y_{t+1} &\sim p_a(\cdot \mid y_t), \\ z_t &\sim p_h(\cdot \mid x_t, y_t), \end{aligned}$$

which capture the noise characteristics. Given a prior pmf $p_t \in \mathcal{P}(\mathcal{Y}) := \{q \in [0,1]^{|\mathcal{Y}|} \mid \sum_{y\in\mathcal{Y}} q(y) = 1\}$ describing the target estimate at time t, since the state and measurement spaces are discrete, we can implement the Bayes filter (Appendix C) for target inference exactly:

$$\textbf{\textit{Predict:}} \qquad p_{t+1|t}(y) = \sum_{j\in\mathcal{Y}} p_a(y \mid j) p_t(j) \tag{4.1}$$

$$\textbf{\textit{Update:}} \qquad p_{t+1}(y) = \frac{p_h(z_{t+1} \mid x_{t+1}, y) p_{t+1|t}(y)}{\sum_{j\in\mathcal{Y}} p_h(z_{t+1} \mid x_{t+1}, j) p_{t+1|t}(j)} \tag{4.2}$$

For notational convenience, let the posterior pmf, obtained via the prediction (4.1) and update (4.2) steps, be denoted $b(p, z, x)$, where p is the prior, $x \in \mathcal{X}$ is the sensor state, and $z \in \mathcal{Z}$ is a new measurement.

Finally, instead of measuring the uncertainty in the target estimate as a performance criterion, since the target state is now discrete, we can directly compute the probability $Pe(t)$ that the maximum-likelihood estimate $\hat{y}_t := \underset{y\in\mathcal{Y}}{\arg\max}\, p_t(y)$ is different from the true target state y_t:

$$Pe(t) := \mathbb{P}(y_t \neq \hat{y}_t) = \mathbb{E}_{z_{1:t}} \sum_{y\in\mathcal{Y}} \left(\mathbb{1}_{\left\{ y \neq \underset{j\in\mathcal{Y}}{\arg\max}\, p_t(j) \right\}} p_t(y) \right) = \mathbb{E}_{z_{1:t}} \left(1 - \max_{j\in\mathcal{Y}} p_t(j) \right). \tag{4.3}$$

This is not a significant change with respect to the conditional entropy criterion $\mathbf{H}(y_t \mid z_{1:t})$ we have been using so far because Fano's inequality (Cover and Thomas 2012) gives a connection between the two criteria:

$$\mathbf{H}(y_t \mid z_{1:t}) \leq Pe(t) \log(|\mathcal{Y}| - 1) + \mathbf{H}(Pe(t)).$$

By minimizing the probability of error, we are again, implicitly, minimizing the entropy of the target state conditioned on the measurement set. In this chapter we also measure the energy expenditure due to the sensor motion via a movement cost $g(x_t, x_{t+1})$. In other words, the objective function aims to balance movement cost and the probability of an incorrect target estimate as described in the problem formulation below.

Problem (Discrete-space Active Information Acquisition)**.** *Given an initial sensor state $x_0 \in \mathcal{X}$, a prior $p_0 \in \mathcal{P}(\mathcal{Y})$ on the true target state y_0, and a finite planning horizon T, choose a sequence of functions $\mu_t(\mathcal{H}_t) \in \mathcal{U}$ for $t = 0, \dots, T-1$, which minimizes the average movement cost and the probability of an incorrect target estimate:*

$$\begin{aligned} \min_{\mu_{0:T-1}} \; & \sum_{t=0}^{T-1} g(x_t, x_{t+1}) + \lambda Pe(T) & & \\ s.t. \; & x_{t+1} = f(x_t, \mu_t(\mathcal{H}_t)), & & t = 0, \dots, T-1, \\ & x_{t+1} \notin \{x_0, \dots, x_t\}, & & t = 0, \dots, T-1, \\ & z_t \sim \sum_{y \in \mathcal{Y}} p_h(\cdot \mid x_t, y) p_t(y), & & t = 1, \dots, T, \\ & p_{t+1} = b(p_t, z_{t+1}, x_{t+1}), & & t = 0, \dots, T-1, \end{aligned} \tag{4.4}$$

where $\lambda \geq 0$ determines the relative importance of a correct estimate versus cost of movement, $Pe(t)$ is the probability of error defined in (4.3)*, and $\mathcal{H}_0 := (x_0, z_0)$, $\mathcal{H}_t := (x_{0:t}, z_{0:t}, u_{0:(t-1)})$, $t > 0$ is the state, measurement, and control history.*

Remark. The second constraint above is optional. It prevents the sensor from revisiting previous states, which is necessary in some applications (e.g., object detection) to ensure that the measurements are independent, as required by the Bayes filter.

4.2 Exact Solution via Dynamic Programming

The active information acquisition problem in (4.4) is an instance of a discrete-space stochastic optimal control problem. As a result, we can use backward value iteration (BVI, Alg. 9) to solve the problem exactly. BVI proceeds backwards in time and first computes the terminal cost (i.e., the probability of error (4.3)) after the last observation z_T has been incorporated in the pmf p_T (Line 3). Afterwards, it computes the intermediate stage costs ($V(x, p, \mathcal{A})$, Line 12) by keeping track of the reachable sensor states (S_t, Line 1) and the admissible control inputs ($\mathcal{U}_t(x)$, Line 2) that ensure that previously visited sensor states are not revisited (second constraint in (4.4)). Unfortunately, the complexity of BVI scales exponentially with the sizes $|\mathcal{X}|$ and $|\mathcal{Y}|$ of the sensor and target state spaces. Nevertheless, Sec. 4.5 shows that the exact algorithm is useful in practical applications such as speeding up single-image object recognition. An alternative to BVI, is to use an approximate POMDP solver such as SARSOP (Kurniawati et al. 2008), which can handle much larger state spaces. However, as discussed earlier, general POMDP solvers do not exploit the properties of our particular cost function, $Pe(t)$, and are typically offline algorithms that have difficulties with some constraints and changing environments. In the next section, we develop an *online* approximate solution to problem (4.4).

Algorithm 9 Backward Value Iteration

```
 1: $S_t := \{S \subseteq \mathcal{X} \mid |S| = t\}$                                     % reachable sensor states at time t
 2: $\mathcal{U}_t(x) := \{s \in \mathcal{X} \setminus S_t \mid s = f(x,u),\ u \in \mathcal{U}\}$                    % admissible controls at time t
 3: $V(x,p,\mathcal{X}) := \lambda\left(1 - \max_{y\in\mathcal{Y}} p(y)\right), \quad \forall x \in \mathcal{X}, p \in \mathcal{P}(\mathcal{Y}) := \{q \in [0,1]^{|\mathcal{Y}|} \mid \sum_{y\in\mathcal{Y}} q(y) = 1\}$
 4:
 5: for $t = T-1,\dots,1$ do
 6:     for $p \in \mathcal{P}(\mathcal{Y})$ do
 7:         for $\mathcal{A} \in S_t$ do
 8:             for $x \in \mathcal{A}$ do
 9:                 for $u \in \mathcal{U}_t(x)$ do
10:                     $s := f(x,u)$
11:                     $Q(x,p,\mathcal{A},u) := g(x,s) + \mathbb{E}_z V\left(s, b(p,z,s), \mathcal{A} \cup \{s\}\right)$
12:                 $V(x,p,\mathcal{A}) := \min_{u\in\mathcal{U}_t(x)} Q(x,p,\mathcal{A},u)$
13:                 $\mu(x,p,\mathcal{A}) := \arg\min_{u\in\mathcal{U}_t(x)} Q(x,p,\mathcal{A},u)$
14: return $\mu$
```

Remark. If the planning horizon T is subject to optimization we get an optimal stopping time problem. The only modification to Alg. 9 that is necessary is to stop the exploration, if at time t, $V(x,p,\mathcal{A}) \geq \lambda(1 - \max_y p(y))$. The inequality can be checked after line 10 in Alg. 9 and if satisfied, $\mu(x,p,\mathcal{A})$ should declare a stopping action. In that case, the decrease in error probability resulting from further observations would not justify the cost of changing the sensor state.

4.3 Approximate Solution via Monte Carlo Tree Search

Monte Carlo tree search (MCTS, Alg. 10) is an online simulation-based alternative to the exact dynamic programming solution of the discrete-space active information acquisition problem (4.4). MCTS constructs a tree sequentially in a best-first order. A node in the tree corresponds to a state (x_t, p_t) and contains a visitation count and the total cost accumulated over all simulations, both initialized to 0. Each simulation has two stages: a *tree policy* and a *rollout policy*. The tree policy (lines 12-18) is followed until reaching a leaf node. The real work of the tree policy is done by the SelectChild function (line 12), which uses the UCT (Upper Confidence bounds applied to Trees) method (Kocsis and Szepesvári 2006) to select the next node as follows:

$$n' = \underset{n\in\textsc{Children}(\text{ node })}{\arg\min} \frac{n.\text{TotalCost}}{n.\text{Visits}} + \kappa\sqrt{\frac{\log(\text{node.Visits})}{n.\text{Visits}}},$$

where κ is an exploration parameter encouraging selection of rarely-visited sensor states. Once a leaf is reached, the rollout policy (lines 9-11) computes the cost-to-go by choosing control inputs and simulating measurements until the end of the planning horizon T. The most common choice for a rollout policy is one that picks successors uniformly at random (Browne et al. 2012).

While the tree policy has been subject to extensive research (Kocsis and Szepesvári 2006, Silver and Veness 2010), since it determines if MCTS converges asymptotically to the optimal policy, the choice of rollout policy has received less attention. MCTS converges for any choice of rollout policy, but the convergence speed can be improved significantly by using domain-specific knowledge to design a rollout policy. Policies with asymptotic optimality

Algorithm 10 Monte Carlo Tree Search

```
 1: root ← $\textsc{TreeNode}(x_0,p_0)$
 2: while within computational budget do
 3:     $\textsc{Simulate}$(root, 0)
 4: best_node = $\underset{n \in \textsc{Children}(\text{root})}{\arg\min} \frac{n.\text{TotalCost}}{n.\text{Visits}}$
 5: return best_node.x
 6:
 7: function $\textsc{Simulate}$(node, $t$)
 8:     if $t = T$ then return $\lambda(1 - \max_y \text{node}.p(y))$
 9:     if $\textsc{IsLeaf}$(node) then
10:         $\textsc{ExpandTree}$(node)
11:         return $\textsc{Rollout}$(node, $t$)
12:     $n' \leftarrow \textsc{SelectChild}(\text{node})$
13:     $z \leftarrow \textsc{SampleObservation}(n'.x,\ p)$
14:     $n'.p \leftarrow b(p, z, n'.x)$
15:     $J \leftarrow g(\text{node}.x, n'.x) + \textsc{Simulate}(n', t+1)$
16:     node.Visits ← node.Visits +1
17:     node.TotalCost ← node.TotalCost $+J$
18:     return $J$
```

guarantees based on Kullback-Leibler divergence (Naghshvar and Javidi 2013b), Jensen-Shannon divergence (Naghshvar et al. 2012) and Chernoff information (Nitinawarat and Veeravalli 2013b, Nitinawarat et al. 2013) have been proposed for active hypothesis testing. Since the discrete-space information acquisition problem (4.4) is closely related to hypothesis testing, these policies are excellent candidates for rollout (Alg. 10, line 11). Instead of the commonly-used uniform policy, we propose applying a rollout policy based on the Extrinsic Jensen-Shannon (EJS) divergence among the observation models corresponding to different sensor states. In particular, at time t, $\mu_t(p_t, x_{0:t})$ selects sensor state $x \in \mathcal{X} \setminus \{x_{0:t}\}$ at random with probability proportional to:

$$\frac{1}{g(x_t, x)} EJS(p_t; \{p_h(\cdot \mid x, y) \mid y \in \mathcal{Y}\}),$$

where EJS is defined below.

Extrinsic Jensen-Shannon (EJS) Divergence (Naghshvar et al. 2012). *The EJS divergence among probability density functions $q_1, \ldots, q_M$ with respect to $p \in [0,1]^M$ is defined as:*

$$EJS(p; q_1, \ldots, q_M) := \sum_{i=1}^{M} p(i) D\left(q_i \middle\| \sum_{k \neq i} \frac{p(k)}{1 - p(i)} q_k\right),$$

where $D(\cdot||\cdot)$ is the Kullback-Leibler divergence (Appendix A).

Intuitively, the proposed rollout policy tries to select a control input that will result in maximum differentiation (measured by the EJS divergence) among the observation models associated with different possible target states. The policy has performance guarantees for active hypothesis testing (Naghshvar and Javidi 2013b) and Bayesian active learning (Naghshvar et al. 2013).

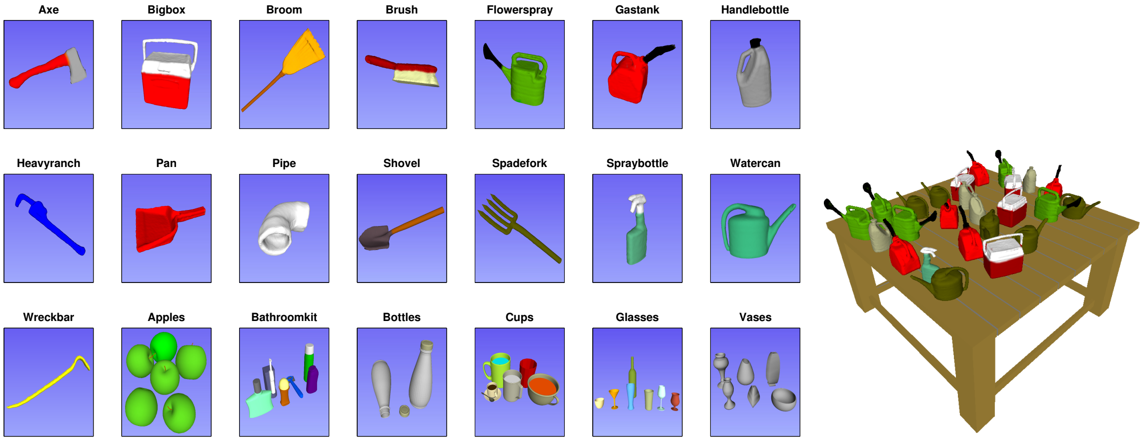


Figure 4.1: Database of object models constructed using kinect fusion (Newcombe et al. 2011) and an example of a scene used to evaluate our framework in simulation.

4.4 Application: Object Classification and Pose Estimation

Consider a mobile depth sensor, which observes a *static* scene, containing unknown objects. The sensor has access to a finite database $\mathcal{D}$ of object models (Fig. 4.1) and a subset $\mathcal{I}$ of them are designated as objects of interest. We assume that an object class has a single model associated with it and use the words model and class interchangeably[2]. The task of the sensor is to detect all objects from $\mathcal{I}$ which are present in the scene and estimate their poses. Note that the detection is not only against known objects from the database but also clutter, background, and unknown objects. At each time step, the sensor obtains a point cloud from the scene, splits it into separate surfaces (*segmentation*) and associates them with either new or previously seen objects (*data association*). These procedures are described in Sec. 4.4.3 and we assume they estimate the object positions accurately.

Hypotheses are formulated about the class and orientation of an unknown object by choosing a *small* finite set of discrete orientations $\mathcal{R}(c) \subset SO(3)$ for each object class $c \in \mathcal{I}$ (see Fig. 4.2). To denote the possibility that an object is not of interest we introduce a dummy class $c_\emptyset$ and a dummy orientation $\mathcal{R}(c_\emptyset) = \{r_\emptyset\}$. The sensor needs to decide among the following hypotheses:

$$\begin{aligned} H(c_\emptyset, r_\emptyset) &: \text{the object does } \textit{not} \text{ belong to } \mathcal{I}, \\ H(c, r) &: \text{the object's class is } c \in \mathcal{I} \text{ with orientation } r \in \mathcal{R}(c) \end{aligned}$$

As described in Sec. 4.1, we maintain a pmf over the hypotheses associated with each object in the scene:

$$p_t(c, r) := \mathbb{P}(H(c, r) \text{ is true} \mid z_{1:t}),$$

where $z_{1:t}$ are the measurements obtained by the camera. Thus, the maximum-likelihood hypothesis at time t is $H(\hat{c}_t, \hat{r}_t)$, where

$$(\hat{c}_t, \hat{r}_t) := \operatorname*{arg\,max}_{c \in \bar{\mathcal{I}}, \hat{r} \in \mathcal{R}(\hat{c})} p_t(c, r)$$

[2] This is necessary because our static detector (Sec. 4.4.1) works with instances. However, the view planning approach is independent of the particular detector and can be used with class-based detectors.

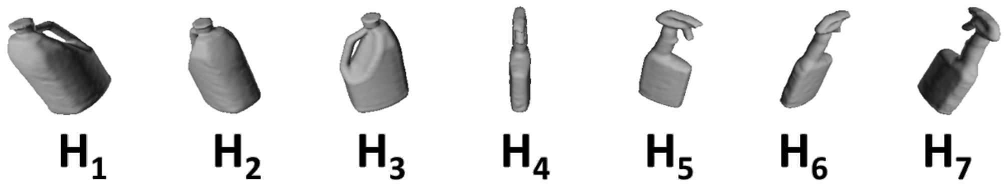


Figure 4.2: An example set of hypotheses about the class and orientation of an unknown object

and $\bar{\mathcal{I}} := \mathcal{I} \cup \{c_\emptyset\}$ denotes the set of all hypothesized classes. To measure the correctness of the sensor's decisions we introduce a cost for choosing $H(\hat{c}_t, \hat{r}_t)$, when $H(c, r)$ is correct:

$$\gamma(\hat{c}, \hat{r}, c, r) := \begin{cases} K(\hat{r}, r), & \hat{c} = c \\ K_+, & \hat{c} \in \mathcal{I}, c \notin \mathcal{I} \\ K_-, & \hat{c} \neq c, c \in \mathcal{I}, \end{cases}$$

where K_+ and K_- are costs for making false positive and false negative mistakes respectively, and $K(\cdot, \cdot)$ is a cost for an incorrect orientation estimate when the class is correct.

Example 4.1. Suppose that the task is to look for chairs (c_1) and tables (c_2) regardless of orientation ($K(\hat{r}, r) := 0$). The decision cost can be represented by the matrix:

$\gamma(\hat{c}, \hat{r}, c, r)$:

$\hat{c}$ \ c	$c_\emptyset$	c_1	c_2
$c_\emptyset$	0	K_-	K_-
c_1	K_+	0	K_-
c_2	K_+	K_-	0

Given p_t, the expected cost of deciding an incorrect hypothesis is

$$J_D(t) := \mathbb{E}_{c,r}\gamma(\hat{c}_t, \hat{r}_t, c, r) = \sum_{c \in \bar{\mathcal{I}}} \sum_{r \in \mathcal{R}(c)} \gamma(\hat{c}_t, \hat{r}_t, c, r) p_t(c, r),$$

which is a weighted version of the probability of error in (4.3).

Further, we restrict the motion of the camera to a set of viewpoints on a sphere of radius ρ, centered at the location of the object (see Fig. 4.3). The camera's orientation is fixed so that it points at the centroid of the object. We denote the space of possible sensor poses by $V(\rho)$ and refer to it as a *viewsphere*. The viewsphere is discretized into a set of viewpoints $\mathcal{X}(\rho)$ and the camera mobility is described by graph with the viewpoints as nodes. Edges in the graph connect nodes which are reachable within a single time step from the current location based on the kinematic restrictions of the camera. In this application, we allow revisiting viewpoints (i.e., we do not include the second constraint in (4.4)) but add a fixed sensing cost to prevent the camera from obtaining an infinite number of measurements without moving. Since the motion graph is known a priori the Floyd-Warshall algorithm can be used to precompute the all-pairs movement cost between viewpoints:

$$g(x, x') := g_M(x, x') + g_0,$$

where $g_M(x, x')$ is the cost of moving from x to x' on the viewsphere $\mathcal{X}(\rho)$ and $g_0 > 0$ is the cost of taking another observation. In the experiments, we use the great-circle distance between two poses $x, x' \in \mathcal{X}(\rho)$ as the mobility cost and $g_0 = 1$ as the measurement

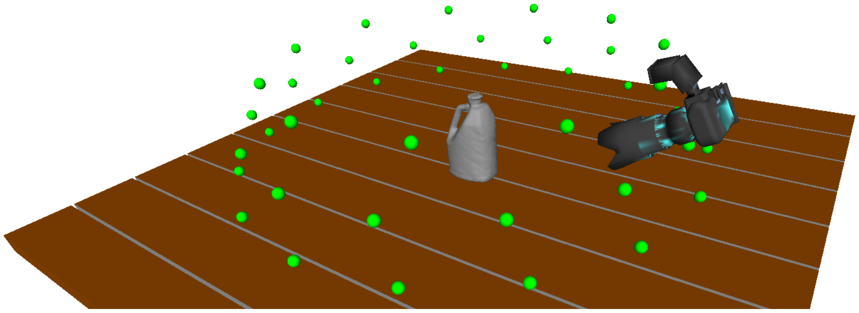


Figure 4.3: Setup for the active object recognition problem. The camera position is restricted to a set of viewpoints (green) on a sphere centered at the object's location. The task is to choose a camera control policy, which minimizes the movement cost and the probability of misclassification.

cost. Thus, a motion plan of length T for the sensor consists of a sequence of viewpoints $x_0, x_1, \ldots, x_T \in \mathcal{X}(\rho)$ on the graph and its cost is $J_M(T) := \sum_{t=0}^{T-1} g(x_t, x_{t+1})$. To summarize, we consider the following problem.

Problem. *Given an initial camera pose $x_0 \in \mathcal{X}(\rho)$ and a prior p_0 over the object hypotheses, choose a stopping time τ, and a sequence of functions μ_t for $t = 0, \ldots, \tau$ to minimize the total cost:*

$$
\begin{aligned}
&\min_{\tau, \mu_{0:\tau}} \mathbb{E}_{z_{1:\tau}} \left[J_M(\tau) + \lambda J_D(\tau) \right] \\
&\quad s.t.\ x_{t+1} = \mu_t, && t = 0, \ldots, \tau - 1, \\
&\qquad z_t \sim \sum_{c,r} p_h(\cdot \mid x_t, c, r) p_t(c, r), && t = 1, \ldots, \tau, \\
&\qquad p_{t+1} = b(p_t, z_{t+1}, x_{t+1}), && t = 0, \ldots, \tau - 1,
\end{aligned}
$$

where, as before, $\lambda > 0$ determines the relative importance of a correct decision versus cost of movement, b is the Bayesian update defined in (4.1) *and* (4.2)*, and $p_h(\cdot \mid x, c, r)$ is the camera observation model.*

First, note that even though the scene contains multiple objects, we formulated the problem with respect to a single object. In other words, we choose a sequence of camera views which are beneficial for a particular object but still update the hypotheses pmfs associated with all other objects in the scene. Applying the approximate algorithm of Sec. 4.3 could handle more objects simultaneously but in this section we use an approximate POMDP solver, which requires a sufficiently small target state space.

Second, note that in addition to computing a camera control policy, the problem above requires optimizing the stopping time. This, however, does not complicate the problem significantly as noted at the end of Sec. 4.2. Finally, note that the only ingredient we did not specify yet is the sensor observation model p_h. This is the subject of the following sections.

4.4.1 Object Detection via the Viewpoint-pose Tree

In this section we introduce a 3D object detector, the viewpoint-pose tree (VP-tree), which provides coarse pose estimates in addition to recognizing an object's class. The VP-Tree is

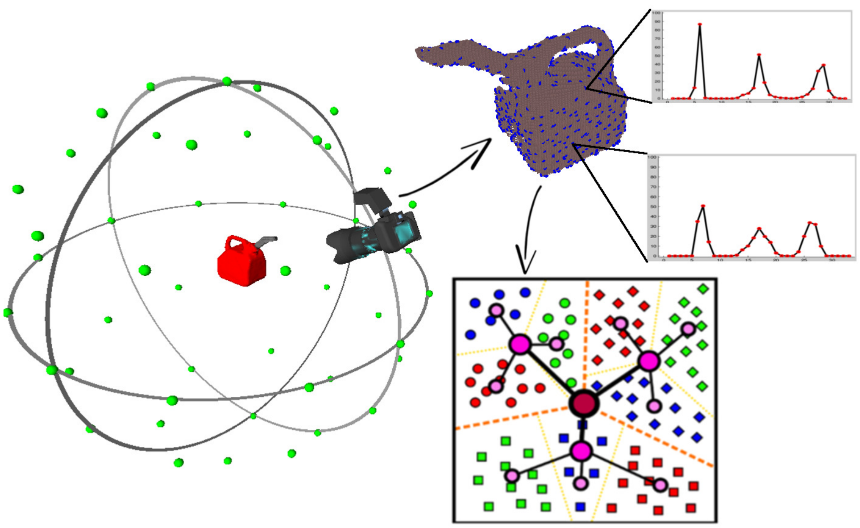


Figure 4.4: The sensor position is restricted to a set of points on a sphere centered at the location of the object. Its orientation is fixed so that it points at the centroid of the object. A point cloud is obtained at each viewpoint, key points are selected, and local features are extracted (top right). The features are used to construct a VP-Tree (bottom right).

built on the principles of the vocabulary tree, introduced by Nistér and Stewénius (2006). A vocabulary tree is primarily used for large scale image retrieval where the number of semantic classes is in the order of a few thousand. The VP-Tree extends the utility of the vocabulary tree to joint recognition and pose estimation in 3D by using point cloud templates extracted from views on a sphere around the models in the database $\mathcal{D}$. The templates serve to discretize the orientation of an object and make it implicit in the detection. Given a query point cloud, the best matching template carries information about both the class and the pose of the object relative to the sensor.

A simulated depth sensor is used to extract templates from a model by observing it from a discrete set $\{v_1(\rho), \ldots, v_G(\rho)\} \subset V(\rho)$ of viewpoints (Fig. 4.4), which need not be the same as the set of planning viewpoints $\mathcal{X}(\rho)$. Here G stands for the number of viewpoints (48 were used in the experiments). The obtained point clouds are collected in a training set $\mathcal{T} := \{\mathcal{P}_{g,l} \mid g = 1, \ldots, G, l = 1, \ldots, |\mathcal{D}|\}$. Features, which describe the local surface curvature are extracted for each template as described below and are used to train the VP-Tree. Given a query point cloud at test time, features are extracted and the VP-Tree is used to find the template from $\mathcal{T}$, whose features match those of the query the closest.

Feature extraction

It is necessary to identify a set of keypoints $\mathcal{K}_{\mathcal{P}}$ for each template $\mathcal{P} \in \mathcal{T}$, at which to compute local surface features. Most 3D features are some variation of surface normal estimation and are very sensitive to noise. Using a unique keypoint estimator would be prone to errors. Instead, the keypoints are obtained by sampling the point cloud uniformly (Fig. 4.4), which accounts for global appearance and reduces noise sensitivity. Neighboring points within a fixed radius of every keypoint are used to compute Fast Point Feature Histograms (Rusu 2009). The features are filtered using a pass-through filter and are assembled in the set $\{f\}_{kp}$ associated with $kp \in \mathcal{K}_{\mathcal{P}}$.

Training the VP-Tree

The features $\bigcup_{\mathcal{P}\in\mathcal{T}} \bigcup_{kp\in\mathcal{K}_{\mathcal{P}}} \{f\}_{kp}$ obtained from the training set are quantized hierarchically into visual words, which are defined by k-means clustering (see Nistér and Stewénius (2006) for more details). Instead of performing unsupervised clustering, the initial cluster centers are associated with one feature from each of the models in $\mathcal{D}$. The training set $\mathcal{T}$ is partitioned into $|\mathcal{D}|$ groups, each consisting of the features closest to a particular cluster center. The same process is applied to each group of features, recursively defining quantization cells by splitting each cell into $|\mathcal{D}|$ new parts. The tree is determined level by level, up to a prespecified maximum number of levels.

Given a query point cloud $\mathcal{Q}$ at test time, we determine its similarity to a template $\mathcal{P}$ by comparing the paths of their features down the VP-Tree. The relevance of a feature at node i is determined by a weight $w_i := \ln\big(|\mathcal{T}|/\eta_i\big)$, where η_i is the number of templates from $\mathcal{T}$ with at least one feature path through node i. The weights are used to define a query descriptor q and a template descriptor $d_{\mathcal{P}}$, with i-th components $q_i := n_i w_i$ and $d_i := m_i w_i$ respectively, where n_i and m_i are the number of features of the query and the template with a path through node i. The templates from $\mathcal{T}$ are ranked according to a relevance score:

$$s(q, d_{\mathcal{P}}) := \left\| \frac{d_{\mathcal{P}}}{\|d_{\mathcal{P}}\|_1} - \frac{q}{\|q\|_1} \right\|_1 .$$

The template with the lowest relevance score is the best matching one to $\mathcal{Q}$.

Performance of the VP-Tree

The performance of the detector was evaluated by using the templates from $\mathcal{T}$ as queries to construct a confusion matrix (Fig. 4.5). If the retrieved template matched the class of the query it was considered correct regardless of the viewpoint. To analyze the noise

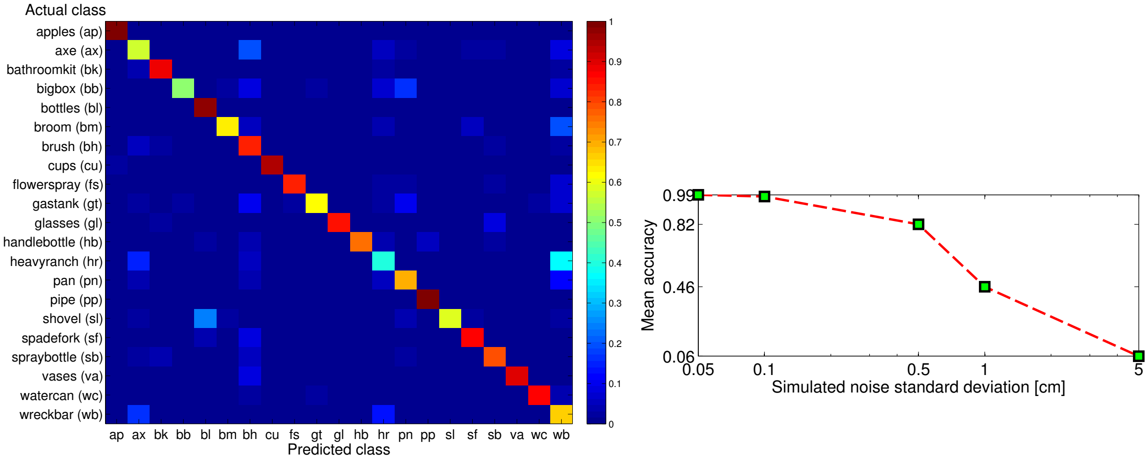


Figure 4.5: Confusion matrix (left) for all classes in the VP-Tree. A class is formed from all views associated with an object. The effect of signal noise on the classification accuracy of the VP-Tree is shown on the right.

sensitivity of the VP-Tree, we gradually increased the noise added to the test set. Gaussian noise with standard deviation varying from 0.05 to 5 cm on a log scale was added along the direction of the ray cast from the observer's viewpoint. As expected the performance starts to degrade as the amount of noise is increased but the detector behaves well at the typical depth camera noise levels.

4.4.2 Camera Observation Model

Statistics about the operation of the VP-Tree detector for different viewpoints $x \in \mathcal{X}(\rho)$ and object classes $c \in \bar{\mathcal{I}}$ and orientations $r \in \mathcal{R}(c)$ are needed to specify the sensor observation model $p_h(\cdot \mid x, c, r)$. Using the VP-Tree output as the sensor observation reduces the observation space from all possible point clouds to the space of VP-Tree outputs and includes the operation of the vision algorithm in the statistics. Given a query point cloud suppose that the VP-Tree returns template $\mathcal{P}_{g,l}$ as the top match. Assume that the templates in $\mathcal{T}$ are indexed so that those obtained from models in $\mathcal{I}$ have a lower l index than the rest. We take the linear index of $\mathcal{P}_{g,l}$ as the observation if the match is an object of interest. Otherwise, we record only the model index l, ignoring the viewpoint g:

$$z = \begin{cases} (l-1)G + g, & \text{if } l \leq |\mathcal{I}| \\ G|\mathcal{I}| + (l - |\mathcal{I}|), & \text{if } l > |\mathcal{I}|. \end{cases}$$

This makes the observation space one dimensional.

In order to compute the likelihood of an observation offline, we introduce an occlusion state ψ for a point cloud. Suppose that the z-axis in the sensor frame measures depth and the xy-plane is the image plane. Given parameters ϵ and $\mathcal{E}$, we say that a point cloud is occluded from left if it has less than $\mathcal{E}$ points in the image plane to the left of the line $x = -\epsilon$. If it has less than $\mathcal{E}$ points in the image plane above the line $y = \epsilon$, it is occluded from top. Similarly, we define occluded from bottom, occluded from right, and combinations of them (left-right, left-top, etc.). Let Ψ denote the set of occlusion states, including the non-occluded ($\psi_\emptyset$) and the fully-occluded cases. Then, the likelihood of a VP-Tree observation z for a given sensor pose $x \in \mathcal{X}(\rho)$, hypothesis $H(c, r)$, and occlusion $\psi \in \Psi$ is $p_h(z \mid x, c, r, \psi)$. The function p_h can be obtained offline because for a given occlusion state it only depends on the characteristics of the sensor and the vision algorithm. Since all variables are discrete, p_h can be represented with a histogram, which we compute from the training set $\mathcal{T}$. Note, however, that the observation model depends on the choice of planning viewpoints and hypotheses, which means that it needs to be recomputed if they change. Ideally, it should be computed once for a given training set and then be able to handle scenarios with different sets of hypotheses and planning viewpoints.

To make the computation of the observation model independent of the choice of hypotheses and planning views we discretize the viewsphere $V(\rho)$ very finely into a new set of viewpoints $V^o(\rho)$ with coordinates in the object frame. A nominal observation model:

$$p_h^o(z \mid v, c, \psi) := \mathbb{P}(Z = z \mid v, c, \psi), \quad v \in V^o(\rho), c \in \mathcal{D}, \psi \in \Psi$$

is computed and used to obtain p_h as follows:

1. Determine the sensor's pose $w(x, r)$ in the object frame.

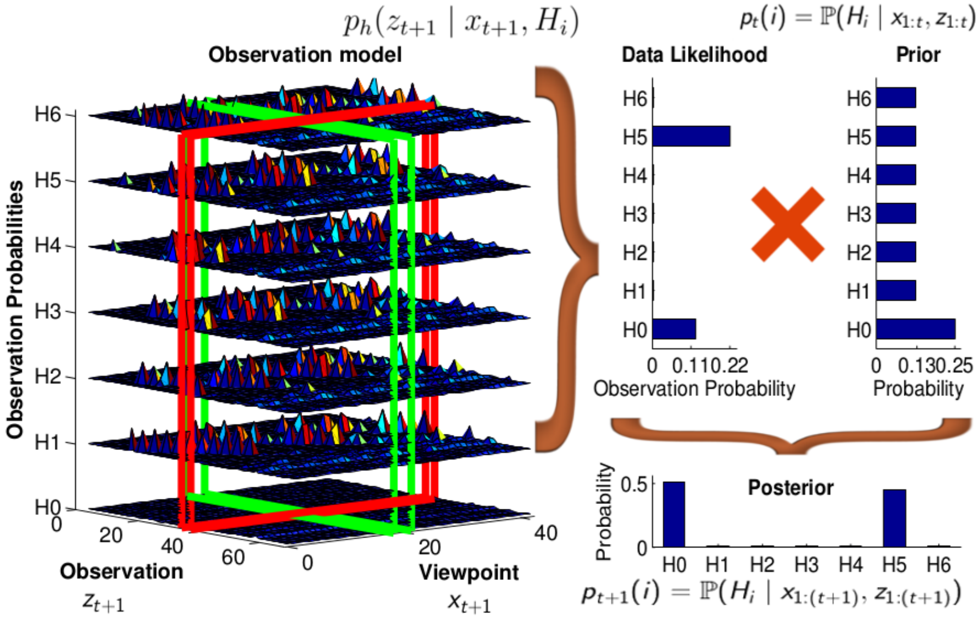


Figure 4.6: Observation model obtained with seven hypotheses for the Handlebottle model and the planning viewpoints used in the simulation experiments (Sec. 4.4.4). Given a new VP-Tree observation z_{t+1} from the viewpoint x_{t+1}, the observation model is used to determine the data likelihood of the observation and to update the hypotheses' prior by applying Bayes rule.

2. Find the closest viewpoint $v \in V^o(\rho)$ to $w(x, r)$ (the fine discretization avoids a large error).

3. Rotate the lines associated with ψ to the object frame of c to get the new occlusion region. Obtain a point cloud from v, remove the points within the occlusion region, and determine the occlusion state ψ^o in the object frame.

4. Copy the values from the nominal observation model:
$p_h(z \mid x, c, r, \psi) := p_h^o(z \mid v, c, \psi^o)$

As a result, it is necessary to compute only the nominal model $p_h^o(z \mid v, c, \psi^o)$. The histogram representing p_h^o was obtained in simulation. A viewsphere with radius $\rho = 1$ m was discretized uniformly into 128 viewpoints (the set $V^o(\rho)$). A simulated depth sensor was used to obtain 20 independent scores from the VP-Tree for every viewpoint $v \in V^o(\rho)$, every model $c \in \mathcal{D}$, and every occlusion state $\psi \in \Psi$. Fig. 4.6 shows an example of the final observation model obtained from the nominal one with the planning viewpoints and hypotheses used in some of our experiments.

4.4.3 Implementation Details

Given the observation model p_h developed in the previous section, we can solve the active object recognition problem via dynamic programming as described in Sec. 4.2. However, instead of the exact backward value iteration algorithm, we apply an approximate point-based POMDP algorithm (SARSOP, Kurniawati et al. (2008)), which uses control and observation samples to compute successive approximations to the optimally reachable part of the continuous space of probability mass functions over the hypotheses. Details can be found in Atanasov et al. (2014b). SARSOP computes a control policy which maps the current camera viewpoint x_t and the hypotheses' probabilities p_t to either a future viewpoint or to a guess of the correct hypothesis in case that the stopping time is reached. The advantage of SARSOP over backward value iteration is that it alleviates the exponential complexity in the number of object hypotheses and the number of camera viewpoints $|\mathcal{X}|$. In practice, there is some control over the number of hypotheses because typically few objects are of interest and we show in Sec. 4.4.4 that a very sparse discretization of the orientation space is sufficient to obtain accurate orientation estimates.

Segmentation and Data Association

To validate the proposed active object recognition approach, we carried out experiments in a tabletop setting. This simplifies the problems of segmentation and data association, which were not the focus of the experiments. Point clouds obtained from the scene were clustered according to Euclidean distance by a k-d tree. An occupancy grid representing the 2D table surface was maintained in order to associate the clustered surfaces with new or previously seen objects. Each cell of the grid could be unoccupied or associated with the ID of an existing object. The centroid of a newly obtained object surface was projected to the table and compared with the occupied cells (if any). If the cell corresponding to the new centroid was close enough to a cell associated with an existing object, the new surface was associated with that object and its cell was indexed by the existing object's ID. Otherwise, a new object with a unique ID was instantiated.

Coupling among Objects

As mentioned earlier, the optimization in problem is with respect to a single object but while executing it, the sensor obtains surfaces from other objects within its field of view. To utilize these observations, we had the sensor turn towards the centroid of every visible object and update the probabilities of the hypotheses associated with the object. The turning is required because the observation model was trained only for a sensor facing the centroid of the object. Removing this assumption requires more training data and complicates the observation model computation. The energy used for these turns was not included in the optimization.

The scores obtained from the VP-Tree are not affected significantly by scaling. This allowed us to vary the radius ρ of the viewsphere in order to ease the sensor movement and to update hypotheses for other objects within the field of view. The radius was set to 1 meter by default but if the next viewpoint was not reachable, it could be adapted to accommodate for obstacles and the sensor dynamics. Algorithm 11 summarizes the complete view planning framework.

4.4.4 Performance Evaluation

The VP-Tree was trained on templates extracted using a simulated depth sensor from 48 viewpoints, uniformly distributed on a viewsphere of radius $\rho = 1$ m (Fig. 4.4). The observation model was trained as described in the last paragraph of Sec. 4.4.2. We used

Algorithm 11 View Planning for Active Object Recognition

```
 1: **Input**: Initial sensor pose $x_0 = (x_0^p, x_0^r) \in \mathbb{R}^3 \ltimes SO(3)$, hypotheses prior $p_0$, object models of interest $\mathcal{I}$
 2: **Output**: Decision $\hat{c}^i \in \bar{\mathcal{I}}, \hat{r}^i \in \mathcal{R}(\hat{c}^i)$ for every object $i$ in the scene
 3: Priority queue $pq \leftarrow \emptyset$;     Current object ID $i \leftarrow$ unassigned
 4: for $t = 0$ to $\infty$ do
 5:     Obtain a point cloud $\mathcal{Q}_t$ from $x_t$
 6:     Cluster $\mathcal{Q}_t$ and update the table occupancy grid
 7:     for every undecided object $j$ seen in $\mathcal{Q}_t$ do
 8:         Rotate the sensor so that $x_t^r$ faces the centroid of $j$
 9:         Get viewsphere radius: $\rho \leftarrow \|x_t^p - centroid(j)\|$
10:         Get closest viewpoint: $v^j \leftarrow \arg\min_{v \in \mathcal{X}(\rho)} \|x_t^p - v\|$
11:         Obtain a point cloud $\mathcal{Q}^j$
12:         Get VP-Tree score $z^j$ and occlusion state $\psi^j$ from $\mathcal{Q}^j$
13:         Update probabilities for object $j$ via Bayes rule: $p_{t+1}^j \leftarrow b(p_t^j, z^j, \psi^j, v^j)$
14:         if $j \notin pq$ then
15:             Insert $j$ in $pq$ according to probability $j \in \mathcal{I}$: $1 - p_{t+1}^j(c_\emptyset, r_\emptyset)$
16:     if $i$ is unassigned then
17:         if $pq$ is not empty then
18:             $i \leftarrow pq.pop()$
19:         else                                  ▷ All objects seen so far have been processed.
20:             if whole scene explored then
21:                 break
22:             else
23:                 Move sensor to an unexplored area and start over
24:     $x_{t+1} \leftarrow \mu(v^i, p_{t+1}^i)$
25:     if $x_{t+1}$ is a hypothesis decision $(c, r)$ then
26:         $\hat{c}^i \leftarrow c$, $\hat{r}^i \leftarrow r$, $i \leftarrow$ unassigned, Go to line 19
27:     else
28:         Move sensor to $x_{t+1}$
```

$|\mathcal{X}(\rho)| = 42$ planning viewpoints in the upper hemisphere of the viewsphere to avoid placing the sensor under the table and the following parameters:

$$\lambda = 75 \qquad \gamma(\hat{c}, \hat{r}, c, r) = \begin{cases} 0, & \hat{c} = c \text{ and } \hat{r} = r \\ 1, & \text{otherwise} \end{cases}$$

where λ was set heuristically high in order to favor correct decisions over speed and thus emphasize the advantage of active view planning over static detection.

Performance evaluation in simulation

The Handlebottle model (Fig. 4.1) was used as a single object of interest, i.e., $\mathcal{I} = \{c_H\}$. Keeping the pitch and roll zero, the space of object yaws was discretized into 6 bins to formulate the hypotheses:

$$\begin{aligned} H(\emptyset) := H(c_\emptyset, r_\emptyset) &= \text{The object is } \textit{not} \text{ a Handlebottle} \\ H(r) := H(c_H, r) &= \text{The object is a Handlebottle with yaw} \\ &\qquad r \in \{0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ\}. \end{aligned}$$

Seventy synthetic scenes were generated with 10 true positives for each of the seven hypotheses. The true positive object was placed in the middle of the table, while the rest of the objects served as occluders. Fig. 4.1 shows an example of the scenes used in the simulation.

Four approaches for selecting sequences of views from $\mathcal{X}(\rho)$ were compared. The static approach takes a single measurement from the starting viewpoint and makes a decision based on the output from the VP-Tree. The second approach is our nonmyopic view planning (NVP). The third approach (random) is a random walk on the viewsphere, which avoids revisiting viewpoints. It ranks the viewpoints, which have not been visited yet, according to the great-circle distance from the current viewpoint. Then, it selects a viewpoint at random among the closest ones. The observation model is used to update the hypotheses' probabilities over time. The experiment is terminated when the probability of one hypothesis is above 60%, i.e., $\tau = \inf\{t \geq 0 \mid \exists (c,r) \text{ such that } p_t(c,r) \geq 0.6\}$, and that hypothesis is chosen as the sensor's decision. This stopping rule was chosen empirically so that the random approach makes about the same number of measurements as NVP. This allows us to compare the informativeness of the chosen sensor views. Last approach is the widely-used greedy mutual information (GMI) approach:

$$\begin{aligned}
\mu_{GMI}(x_{0:t}, p_t) &= \operatorname*{arg\,max}_{v \in \mathcal{X}(\rho) \setminus \{x_{0:t}\}} \frac{\mathbf{I}(H(c,r); z_{t+1})}{g(x_t, v)} = \operatorname*{arg\,min}_{v \in \mathcal{X}(\rho) \setminus \{x_{0:t}\}} \frac{\mathbf{H}(H(c,r) \mid z_{t+1})}{g(x_t, v)} \\
&= \operatorname*{arg\,min}_{v \in \mathcal{X}(\rho) \setminus \{x_{0:t}\}} \frac{1}{g(x_t, v)} \sum_{z \in \mathcal{Z}} \sum_{c \in \bar{\mathcal{I}}} \sum_{r \in \mathcal{R}(c)} p_t(c,r) p_h(z \mid v, c, r, \psi_\emptyset) \\
&\qquad \times \log_2 \left(\frac{\sum_{c' \in \bar{\mathcal{I}}} \sum_{r' \in \mathcal{R}(c')} p_t(c', r') p_h(z \mid v, c', r', \psi_\emptyset)}{p_t(c,r) p_h(z \mid v, c, r, \psi_\emptyset)} \right),
\end{aligned}$$

where $H(c,r)$ is the true hypothesis, $\mathbf{I}(\cdot;\cdot)$ is mutual information, and $\mathbf{H}(\cdot \mid \cdot)$ is conditional entropy. The same stopping rule as for the random approach was used so that the number of measurements made by GMI is roughly the same as those for random and NVP.

Fifty repetitions with different starting sensor poses were carried out on every scene. For each hypothesis, the measurement cost $\sum_{t=0}^{\tau-1} g_0$, the movement cost $\sum_{t=0}^{\tau-1} gcd(x_t, x_{t+1})$, and the decision cost J_D were averaged over all repetitions. The accuracy of each approach and the average costs are presented in Table 4.1. The following conclusions can be made:

- The active approaches for object classification and pose estimation significantly outperform the traditional single-view approach in terms of accuracy. In most cases, by making $1-2$ extra measurements, they are able to choose the correct hypothesis more than 20% more frequently.

- There is a steady improvement in performance when going from random viewpoint selection, to greedy view planning, and finally to nonmyopic view planning. Compared with the random and the GMI approaches, our NVP method needs less movement and less measurements on average and, as demonstrated by its lower average decision cost, is able to select more informative views.

- The performance gain of NVP over GMI is not significant. In some scenarios it might not justify the complicated offline training. For example, it is much easier to include additional constraints, such as occlusion avoidance, with greedy planning. This suggests that it is best to use an online nonmyopic closed-loop planning approach, such as the Monte Carlo tree search developed in Sec. 4.3.

Table 4.1: Simulation results for a bottle detection experiment

			True Hypothesis							Avg Number of Measurements	Avg Movement Cost	Avg Decision Cost	Avg Total Cost
			H(0°)	H(60°)	H(120°)	H(180°)	H(240°)	H(300°)	H(∅)				
Predicted Hypothesis (%)	Static	H(0°)	60.35	3.86	1.00	2.19	1.48	2.19	28.92	1.00	0.00	29.74	30.74
		H(60°)	5.53	53.90	2.19	1.00	1.48	1.95	33.94	1.00	0.00	34.57	35.57
		H(120°)	4.86	4.62	51.49	3.90	2.21	1.24	31.68	1.00	0.00	36.38	37.38
		H(180°)	4.34	4.34	6.01	49.13	1.95	1.24	32.98	1.00	0.00	38.15	39.15
		H(240°)	3.88	1.96	1.24	2.20	56.11	1.24	33.37	1.00	0.00	32.92	33.92
		H(300°)	5.07	1.24	2.44	2.44	1.72	54.29	32.82	1.00	0.00	34.28	35.28
		H(∅)	0.56	1.09	3.11	1.93	0.32	3.13	89.87	1.00	0.00	7.60	8.60
												Overall Average Total Cost:	**31.52**
	Random	H(0°)	73.78	3.17	1.24	2.21	1.48	1.24	16.87	2.00	1.26	19.66	22.93
		H(60°)	1.96	70.34	2.20	1.72	1.00	1.48	21.31	2.36	1.71	22.25	26.31
		H(120°)	1.00	1.49	70.75	3.43	1.00	1.24	21.09	2.30	1.64	21.94	25.87
		H(180°)	1.48	1.73	3.66	66.97	1.97	1.48	22.71	2.71	2.16	24.78	29.64
		H(240°)	1.48	1.24	1.48	2.45	68.76	1.72	22.87	2.41	1.77	23.43	27.62
		H(300°)	1.72	1.97	1.00	1.24	1.97	71.85	20.25	2.60	2.02	21.11	25.74
		H(∅)	0.07	2.11	2.00	1.53	1.59	0.37	92.33	4.95	4.93	5.76	15.64
												Overall Average Total Cost:	**24.82**
	Greedy MI	H(0°)	82.63	2.93	0.76	1.61	0.83	0.40	10.85	1.96	1.20	13.03	16.19
		H(60°)	0.80	80.14	1.05	1.07	0.14	1.16	15.64	2.26	1.58	14.89	18.73
		H(120°)	1.09	1.05	76.93	2.64	0.83	0.82	16.66	2.30	1.64	17.31	21.25
		H(180°)	1.47	1.25	3.62	75.60	0.71	0.50	16.84	2.79	2.25	18.30	23.34
		H(240°)	0.49	1.15	0.82	2.58	75.29	1.71	17.96	2.37	1.72	18.53	22.62
		H(300°)	1.79	0.50	0.12	0.86	1.21	81.78	13.74	2.59	2.00	13.66	18.25
		H(∅)	0.72	1.35	2.23	0.39	0.25	0.41	94.65	5.29	5.37	4.01	14.67
												Overall Average Total Cost:	**19.29**
	NVP	H(0°)	87.98	0.48	0.24	0.24	0.24	0.48	10.34	2.06	1.45	9.01	12.51
		H(60°)	0.00	83.78	0.97	0.24	0.24	0.24	14.53	2.28	1.73	12.17	16.17
		H(120°)	0.48	0.00	82.81	1.21	0.00	0.00	15.50	2.37	1.86	12.89	17.12
		H(180°)	0.00	0.00	0.97	82.61	1.21	0.24	14.98	2.50	2.05	13.04	17.60
		H(240°)	0.49	0.24	0.00	0.49	78.73	0.00	20.05	2.57	2.18	15.95	20.71
		H(300°)	0.00	0.24	0.24	0.73	0.48	81.60	16.71	2.60	2.15	13.80	18.55
		H(∅)	1.49	1.58	1.37	0.37	0.74	1.25	93.20	2.08	1.50	5.10	8.68
												Overall Average Total Cost:	**15.91**

- The most notable advantage of NVP comes from the adaptive stopping criterion. This is especially evident when the observed object is clutter ($H(\emptyset)$ is correct). In this case, the scores provided by the VP-Tree are not consistent and cause the probabilities of various hypotheses to increase and decrease frequently. As a result, the GMI and random approaches need many measurements to reach their prespecified stopping time. In contrast, NVP employs a longer planning horizon and recognizes that if the clutter class is likely, it is better to stop sooner than to attempt to increase the confidence as many (costly) measurements would be needed.

Accuracy of the orientation estimates

Since the object orientations in a real scene are not discrete, a refinement step is needed if the algorithm detects an object of interest. The surfaces observed from an object are accumulated over time and, after a decision, are aligned using the iterative closest point algorithm to the surface of the database model, corresponding to $H(\hat{c}, \hat{r})$. Thus, the final decision includes both a class and a continuous pose estimate.

Simulations were carried out to evaluate the accuracy of the continuous orientation estimates with respect to the ground truth. The following distance metric on $SO(3)$ was used to measure the error between two orientations represented by quaternions q_1 and q_2:

$$d(q_1, q_2) := \cos^{-1}\big(2\langle q_1, q_2\rangle^2 - 1\big),$$

where $\langle a_1 + b_1\mathbf{i} + c_1\mathbf{j} + d_1\mathbf{k}, a_2 + b_2\mathbf{i} + c_2\mathbf{j} + d_2\mathbf{k}\rangle = a_1a_2 + b_1b_2 + c_1c_2 + d_1d_2$ denotes the quaternion inner product. A single object of interest (Watercan) was used: $\mathcal{I} = \{c_W\}$.

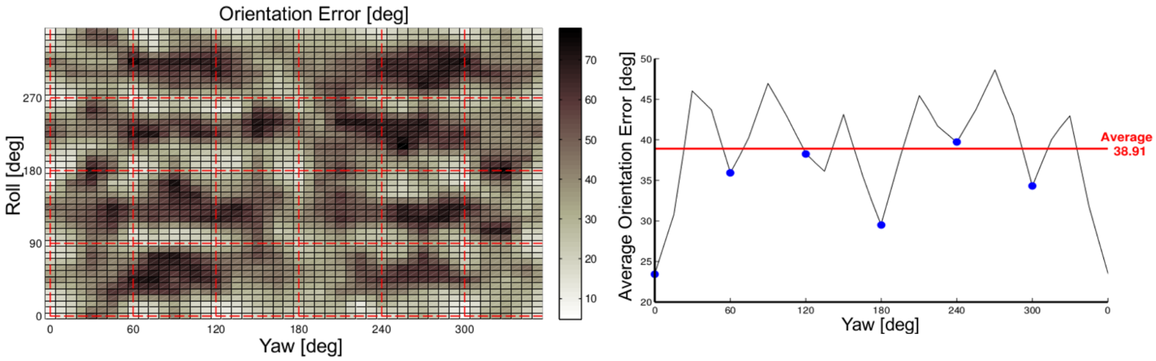


Figure 4.7: Twenty five hypotheses (red dotted lines) were used to decide on the orientation of a Watercan. The left plot shows the error in the orientation estimates as the ground truth orientation varies. The error averaged over the ground truth roll, the hypotheses over the object's yaw (blue dots), and the overall average error (red line) are shown on the right.

The ground truth yaw (α) and roll (γ) of the Watercan were varied from 0° to 360° at 7.5° increments. The pitch (β) was kept zero. Synthetic scenes were generated for each orientation. Hypotheses were formulated by discretizing the yaw space into 6 bins and the roll space into 4 bins:

$$\begin{aligned} H(c_\emptyset, r_\emptyset) =& \text{ The object is } \textit{not} \text{ a Watercan} \\ H(c_W, r) =& \text{ The object is a Watercan with orientation} \\ & \quad r = (\alpha, \beta, \gamma) \in \{(i_y 60^\circ, 0, i_r 90^\circ) \mid i_y = 0, \dots, 5, i_r = 0, \dots, 3\} \end{aligned}$$

Fifty repetitions with different starting sensor poses were carried out on every test scene. The errors in the orientation estimates were averaged and the results are presented in Fig. 4.7. As expected, the orientation estimates get worse for ground truth orientations which are further away from the hypothesized orientations. On the right plot, it can be seen that the hypothesized yaws correspond to local minima in the orientation error. This suggests that the number of hypotheses needs to be increased if a better orientation estimate is desired. Still, a rather sparse set of hypothesized orientations was sufficient to obtain an average error of 39°. For these experiments, the average number of measurements was 2.85 and the average movement cost was 2.61.

Performance evaluation in real-world experiments

In this section, we demonstrate that the real-world performance of NVP is similar to the simulation. We expect the same to be true for the rest of the view planning methods and did not carry out additional real experiments. An Asus Xtion RGB-D camera attached to the right wrist of a PR2 robot was used as the mobile sensor. As before, the sensor's task was to detect if any Handlebottles ($\mathcal{I} = \{c_H\}$) are present on a cluttered table and estimate their poses. Fig. 4.8 shows the experimental setup.

Twelve table setups were used, each containing 2 instances of the object of interest and $8 - 10$ other objects. Ten repetitions were carried out for each setup, which in total corresponded to 40 true positive cases for every hypothesis. The results are summarized in Table 4.2. The performance obtained in the real experiments is comparable to the

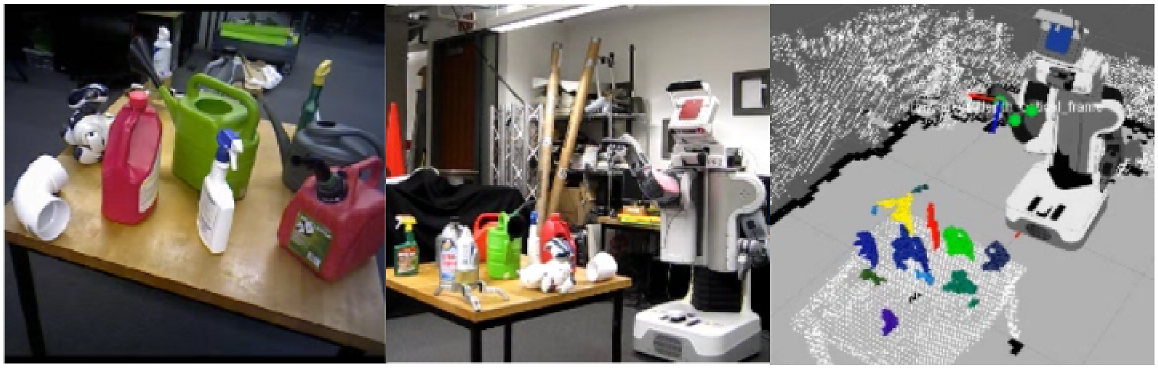


Figure 4.8: An example of the experimental setup (left), which contains two instances of the object of interest (Handlebottle). A PR2 robot with an Asus Xtion RGB-D camera attached to the right wrist (middle) employs the nonmyopic view planning approach for active object classification and pose estimation. In the robot's understanding of the scene (right), the object which is currently under evaluation is colored yellow. Once the system makes a decision about an object, it is colored green if it is of interest, i.e. in $\mathcal{I}$, and red otherwise. Hypothesis $H(0°)$ (Handlebottle with yaw 0°) was chosen correctly for the green object. See the video in Appendix D.16 Extension 16 for more details.

Table 4.2: Results for a real-world bottle detection experiment

		True Hypothesis							Avg Number of Measurements	Avg Movement Cost	Avg Decision Cost	Avg Total Cost
		H(0°)	H(60°)	H(120°)	H(180°)	H(240°)	H(300°)	H(∅)				
Predicted (%)	H(0°)	87.5	2.5	5.0	0.0	0.0	0.0	5.0	2.53	2.81	9.38	14.72
	H(60°)	2.5	80.0	0.0	0.0	0.0	0.0	17.5	2.66	2.52	15.00	20.18
	H(120°)	7.5	0.0	72.5	0.0	0.0	0.0	20.0	3.16	3.43	20.63	27.22
	H(180°)	0.0	0.0	0.0	70.0	10.0	2.5	17.5	2.20	1.72	22.5	26.42
	H(240°)	0.0	0.0	0.0	2.5	75.0	2.5	20.0	2.39	2.51	18.75	23.65
	H(300°)	0.0	0.0	0.0	0.0	5.0	72.5	22.5	2.57	2.18	20.63	25.38
	H(∅)	0.0	0.0	0.97	0.0	0.0	0.97	98.05	2.17	1.93	1.46	5.56
											Overall Average Total Cost:	**20.45**

simulation results. On average, more movement and more measurements were required to make a decision in practice than in simulation, which can be attributed to the fact that the VP-Tree and the observation model were trained in simulation but were used to process real observations. The VP-Tree scores were sometimes inconsistent which caused the hypotheses' probabilities to fluctuate and hence the sensor took longer to make decisions. Still, the results from the experiments are very satisfactory with an average accuracy of 76% for true positives and 98% for true negatives.

There are several aspects of our framework that slow down the processing and need improvement, however. First, the occlusion model should be used in combination with online planning (e.g., the Monte Carlo tree search in Sec. 4.3) to avoid viewpoints with limited visibility. Second, as an artifact of the way the camera observation model was trained, the camera had to turn towards the centroid of every object in its field of view. This is slow and undesirable. The observation model can be modified, at the expense of a more demanding training stage, to include sensor poses which do not face the object's centroid. Finally, an unavoidable computational cost is due to feature extraction from the observed surfaces and to point cloud registration, needed to localize the sensor in the global frame.

4.5 Application: Accelerating Object Recognition with Deformable Part Models

Next, we turn attention to object detection in a single image. Even though there is no sensor to control anymore, we can use active information acquisition to accelerate the evaluation of a star-structured object model such as a deformable part model (Felzenszwalb et al. 2010b). A star-structured model of an object with n parts is formally defined by a $(n+2)$-tuple $(F_0, P_1, \ldots, P_n, \beta)$, where F_0 is a root filter, β is a real-valued bias term, and P_k are the part models. Each part model $P_k = (F_k, v_k, d_k)$ consists of a filter F_k, a position v_k of the part relative to the root, and the coefficients d_k of a quadratic function specifying a deformation cost of placing the part away from v_k. The object detector is applied in a sliding-window fashion to each location q in an image pyramid, where $q = (r, c, l)$ specifies a pixel position (r, c) in the l-th level (scale) of the pyramid. The space of all locations (position-scale tuples) in the image pyramid is denoted by $\mathcal{Q}$. The response of the detector at a given root location $q = (r, c, l) \in \mathcal{Q}$ is:

$$\mathit{score}(q) = F_0' \cdot \phi(H, q) + \sum_{k=1}^{n} \max_{q_k} \Big(F_k' \cdot \phi(H, q_k) - d_k \cdot \phi_d(\delta_k) \Big) + \beta,$$

where $\phi(H, q)$ is the histogram of oriented gradients (HOG) feature vector at location q, H is the HOG feature pyramid, $\delta_k := (r_k, c_k) - (2(r, c) + v_k)$ is the displacement of the k-th part from its anchor position v_k relative to the root location q, and $\phi_d(dr, dc) := (dr, dc, dr^2, dc^2)$ are deformation features. Each term in the sum above implicitly depends on the root location q since the part locations q_k are chosen relative to it. The score can be written as:

$$\mathit{score}(q) = \sum_{k=0}^{n} z_k(q) + \beta, \tag{4.5}$$

where $z_0(q) := F_0' \cdot \phi(H, q)$ and for $k > 0$, $z_k(q) := \max_{q_k} \big(F_k' \cdot \phi(H, q_k) - d_k \cdot \phi_d(\delta_k) \big)$. From this perspective, there is no difference between the root and the parts and we can think of the model as one consisting of $n + 1$ parts.

4.5.1 Observation Model of the Part Filters

The object detection task requires labeling every $q \in \mathcal{Q}$ with a label $y(q) \in \{\ominus, \oplus\}$, indicating if an object is present or not. Since the same task needs to be performed at every location, in the reminder we fix a particular root location $q \in \mathcal{Q}$ and suppress it in the notation. The traditional approach to label y, is to compute the complete score in (4.5) and compare it to a pre-specified threshold. We argue that it is not necessary to obtain all $n + 1$ part responses in order to label a location correctly. Treating the part filter responses as noisy observations of the true label y, we choose an effective order in which to receive observations and an optimal time to stop. The stopping criterion is based on a trade-off between the cost of obtaining more observations and the cost of labeling the location incorrectly.

In the language of active information acquisition, we can think of the true label y as a binary target state, of the part scores $z_0, \ldots, z_n$ as sensor observations, and of the particular part $x_t \in \{0, \ldots, n\}$ to be applied at time t as the sensor state. Before we

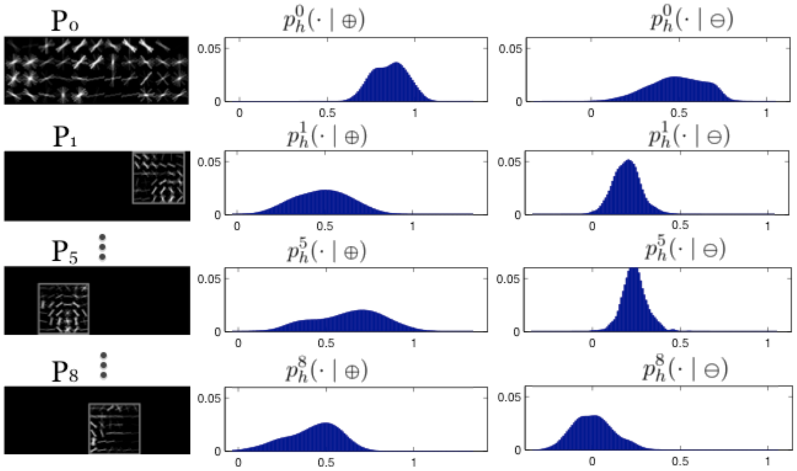


Figure 4.9: Observation models for several part filters from a car DPM. The root (P_0) and three parts of the model are shown on the left. The corresponding positive and negative score likelihoods are shown on the right.

formulate the problem precisely, we present the observation model associated with the filter responses. As the scores $z_0, \ldots, z_n$ are random variables, we need to specify their joint probability density function $p_h(z_0, \ldots, z_n \mid y)$ conditioned on the true label y. We make the following independence assumption, which does not always hold in practice but simplifies the representation of the observation model, avoids overfitting, and allows us to use Bayesian filtering.

Assumption. *The responses of the parts of a star-structured model with a given root location $q \in \mathcal{Q}$ are independent, conditioned on the the true label $y(q)$, i.e.,*

$$p_h(z_0, z_1, \ldots, z_n \mid y) = \prod_{k=0}^{n} p_h(z_k \mid k, y) \tag{4.6}$$

where $p_h(z_k \mid k, \oplus)$ is the pdf of $z_k \mid y = \oplus$ and $p_h(m_k \mid k, \ominus)$ is the pdf of $z_k \mid y = \ominus$.

We learn non-parametric representations for the $2(n+1)$ pdfs $\{p_h(\cdot \mid k, \oplus), p_h(\cdot \mid k, \ominus)\}$ by collecting filter responses for each part from an annotated set D of training images. Given a positive example $I_i^{\oplus} \in D$ of a particular DPM component, the root was placed at the scale and position q^* of the top score within the ground-truth bounding box. The response z_0^i of the root filter was recorded. The parts were placed at their optimal locations relative to the root location q^* and their scores $z_k^i,\ k > 0$ were recorded as well. This procedure was repeated for all positive examples in D to obtain a set of scores $\{z_k^i \mid \oplus\}$ for each part k. For negative examples, q^* was selected randomly over all locations in the image pyramid and the same procedure was used to obtain the set $\{z_k^i \mid \ominus\}$. Kernel density estimation was applied to the score collections in order to obtain smooth approximations to $p_h(\cdot \mid k, \oplus)$ and $p_h(\cdot \mid k, \ominus)$. Fig. 4.9 shows the observation models associated with the part responses of a car model.

4.5.2 Active Part Selection

Now that we have observation models for the filter responses, we can solve a discrete-space active information acquisition problem (4.4) in order to plan an informative order in which to apply the part filters and a time at which to stop and guess the label of $y \in \{\ominus, \oplus\}$. More precisely, starting with an uninformative prior $p_0 = 1/2$ for the true label, we consider the following problem:

$$
\begin{aligned}
\min_{\tau, \mu_{0:\tau-1}} \;& \mathbb{E}[\tau] + \lambda Pe(\tau) \\
\text{s.t. } & x_t = \mu_t, && t = 0, \ldots, \tau - 1, \\
& x_{t+1} \notin \{x_0, \ldots, x_t\}, && t = 0, \ldots, \tau - 2, \\
& z_t \sim \sum_{y \in \{\ominus, \oplus\}} p_h(\cdot \mid x_t, y) p_t(y), && t = 0, \ldots, \tau - 1, \\
& p_{t+1} = b(p_t, z_t, x_t), && t = 0, \ldots, \tau - 1,
\end{aligned}
$$

which is an instance of (4.4) with $g(x_t, x_{t+1}) = 1$ and planning horizon $0 \leq \tau \leq n + 1$ that is subject to optimization. Since the measurement and state spaces are small, we can apply the exact backward value iteration algorithm (Alg. 9) to obtain the optimal control policy $\mu_{0:\tau}$ that maps the set of previously used parts $x_{0:t-1}$ and the pdf p_t to the next filter to be applied. Upon termination, we choose the label $\hat{y}_\tau := \arg\max_{y \in \{\ominus, \oplus\}} p_\tau(y)$ for the current location. The constant λ can be interpreted as a cost paid for choosing an incorrect label. To allow flexibility, we rewrite the cost function as follows:

$$
\mathbb{E}\left[\tau + \lambda \mathbb{E}_y \left[\mathbb{1}_{\{\hat{y}_\tau \neq y\}} \mid z_{0:(\tau-1)}\right]\right] = \mathbb{E}\left[\tau + \sum_{y \in \{\ominus, \oplus\}} \lambda \mathbb{1}_{\{\hat{y}_\tau \neq y\}} p_\tau(y)\right]
$$

and introduce separate costs λ_{fp} and λ_{fn} for false positive and false negative mistakes:

$$
\mathbb{E}\left[\tau + \lambda_{fn} \mathbb{1}_{\{\hat{y}_\tau = \ominus\}} p_\tau(\oplus) + \lambda_{fp} \mathbb{1}_{\{\hat{y}_\tau = \oplus\}} p_\tau(\ominus)\right].
$$

During inference, the policy is used to select the sequence of parts to apply at each location $q \in \mathcal{Q}$ in the image pyramid. Note that the labeling of a different location is treated as an independent problem. The active DPM inference process is summarized in Alg. 12 and exemplified in Fig. 4.10. At the start of a detection at location q, the set of previously used parts s_0 is empty and $p_0 = 1/2$ (line 5). At each round t, the policy is queried to obtain either the next part to run or a predicted label for q (line 7). Note that querying the policy is an $O(1)$ operation since it is stored as a lookup table. If the policy terminates and labels $y(q)$ as foreground (line 8), all unused part filters are applied in order to obtain the final discriminative score in (4.5). On the other hand, if the policy terminates and labels $y(q)$ as background, no additional part filters are evaluated and the final score is set to $-\infty$ (line 16). In this case, our algorithm makes computational savings compared to the original DPM inference. The potential speed-up and the effect on the accuracy are discussed in the Sec. 4.5.3. Finally, if the policy returns a part index k, the corresponding score $z_k(q)$ is computed by applying the part filter (line 19). This operation

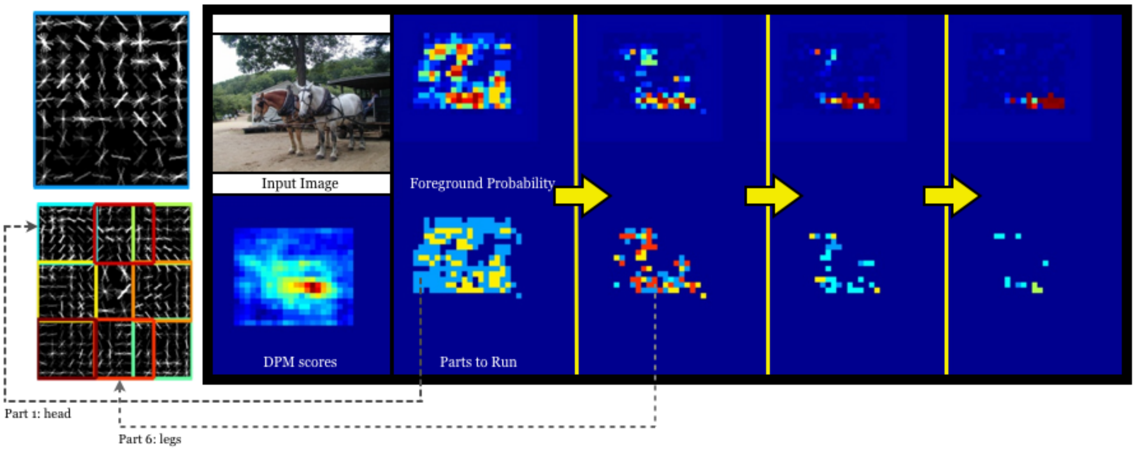


Figure 4.10: **Active DPM Inference**: A deformable part model trained on the PASCAL VOC 2007 horse class is shown with colored root and parts in the first column. The second column contains an input image and the original DPM scores as a baseline. The rest of the columns illustrate the ADPM inference which proceeds in rounds. The foreground probability of a horse being present is maintained at each image location (top row) and is updated sequentially based on the responses of the part filters (high values are red; low values are blue). A policy (learned offline) is used to select the best sequence of parts to apply at different locations. The bottom row shows the part filters applied at consecutive rounds with colors corresponding to the parts on the left. The policy decides to stop the inference at each location based on the confidence of foreground. As a result, the complete sequence of part filters is evaluated at very few locations, leading to a significant speed-up versus the traditional DPM inference. Our experiments show that the accuracy remains unaffected.

is $O(|\Delta|)$, where Δ is the space of possible displacements for part k with respect to the root location q. Following the analysis in Felzenszwalb et al. (2010a), searching over the possible locations for part k is usually no more expensive than evaluating its linear filter F_k once. This is the case because once F_k is applied at some location q_k, the resulting response $\Phi_k(q_k) = F_k' \cdot \phi(H, q_k)$ is cached to avoid recomputing it later. The score z_k of part k is used to update the total score at q (line 20). Since the policy lookups and the state updates are all of $O(1)$ complexity, the worst-case complexity of Alg. 12 is $O(n|\mathcal{Q}||\Delta|)$. The average running time of our algorithm depends on the total number of score z_k evaluations, which in turn depends on the choice of the parameters λ_{fn} and λ_{fp} and is the subject of the next section.

4.5.3 Parameter Selection

The accuracy and speed of active DPM (ADPM) inference depend on the costs λ_{fp} and λ_{fn}. To get an intuition, consider making both λ_{fp} and λ_{fn} very small. The cost of an incorrect prediction will be negligible, thus encouraging the policy to sacrifice accuracy and stop immediately. On the other extreme, when both parameters are very large, the policy will delay the prediction as much as possible in order to obtain more information.

To evaluate the effect of different parameters, we compared the average precision (AP) and the number of part evaluations of Alg. 12 to those of the traditional DPM as a baseline. Let R_M be the total number of score $z_k(q)$ evaluations for $k > 0$ (excluding the root) over all locations $q \in \mathcal{Q}$ performed by method M. For example, $R_{DPM} = n|\mathcal{Q}|$ since the DPM

Algorithm 12 Active DPM Inference

```
Input: Image pyramid, model $(F_0, P_1, \dots, P_n, \beta)$, observation models $\{p_h(\cdot \mid k, \ominus), p_h(\cdot \mid k, \oplus)\}_{k=0}^{n}$, policy $\mu_{0:\tau}$
Output: $score(q)$ at all locations $q \in \mathcal{Q}$ in the image pyramid

for $q \in 1 \dots |\mathcal{Q}|$ do                                        ▷ All image pyramid locations
    $x_0 := -1$; $s_0 := \emptyset$; $p_0 = 0.5$; $score(q) := 0$
    for $t = 0, 1, \dots, n$ do
        $k := \mu_t(x_t, s_t, p_t)$                                          ▷ Lookup next best part
        if $k = \oplus$ then                                                 ▷ Terminate with foreground label
            for $i \in \{0, 1, \dots, n\}$ do
                if $i \notin s_t$ then
                    Compute score $z_i(q)$ for part $i$                      ▷ $O(|\Delta|)$
                    $score(q) := score(q) + z_i(q)$
            $score(q) := score(q) + \beta$                                   ▷ Add bias to final score
            break;
        else if $k = \ominus$ then                                           ▷ Terminate with background labeled
            $score(q) := -\infty$
            break;
        else                                                                 ▷ Update probability and score
            Compute score $z_k(q)$ for part $k$                              ▷ $O(|\Delta|)$
            $score(q) := score(q) + z_k(q)$
            $p_{t+1} := b(p_t, z_k, k)$                                      ▷ Bayesian update
            $s_{t+1} := s_t \cup \{k\}$                                      ▷ Set of used parts
            $x_{t+1} := k$
```

Table 4.3: Average precision and relative number of part evaluations versus DPM obtained on the bus class from VOC 2007 *training* set. A grid search over $(\lambda_{fp}, \lambda_{fn}) \in \{4, 8, \dots, 64\} \times \{4, 8, \dots, 64\}$ with $\lambda_{fp} \geq \lambda_{fn}$ is shown.

Average Precision						RNPE vs DPM					
$\lambda_{fp}/\lambda_{fn}$	4	8	16	32	64	$\lambda_{fp}/\lambda_{fn}$	4	8	16	32	64
4	70.3					4	40.4				
8	70.0	71.0				8	80.7	61.5			
16	69.6	71.1	71.5			16	118.6	74.5	55.9		
32	70.5	70.7	71.6	71.6		32	178.3	82.1	59.8	37.0	
64	67.3	69.6	71.5	71.6	71.4	64	186.9	96.4	56.2	34.5	20.8

evaluates all parts at all locations in $\mathcal{Q}$. We define the *relative number of part evaluations* (RNPE) of ADPM versus method M as the ratio of R_M to R_{ADPM}. The AP and the RNPE versus DPM of ADPM were evaluated on several classes from the PASCAL VOC 2007 training set (Everingham et al. 2010) for different values of the parameter $\lambda = \lambda_{fn} = \lambda_{fp}$ (see Fig. 4.11). As expected, the AP increases while the RNPE decreases, as the penalty of an incorrect declaration λ grows, because ADPM evaluates more parts. The dip in RNPE for very low λ is due to fact that ADPM starts reporting many false-positives. In the case of a positive declaration all $n + 1$ part responses need to be computed which reduces the acceleration versus DPM.

To limit the number of false positive mistakes made by the policy we set $\lambda_{fp} > \lambda_{fn}$. While this might hurt the accuracy, it certainly results in less positive declarations and in turn significantly less part evaluations. To verify this intuition, we performed experiments with $\lambda_{fp} > \lambda_{fn}$ on the VOC 2007 training set. Table 4.3 reports the AP and the RNPE versus DPM from a grid search over the parameter space. Generally, as the ratio between λ_{fp} and λ_{fn} increases, the RNPE increases while the AP decreases. Notice, however, that the increase in RNPE is significant, while the hit in accuracy is negligible.

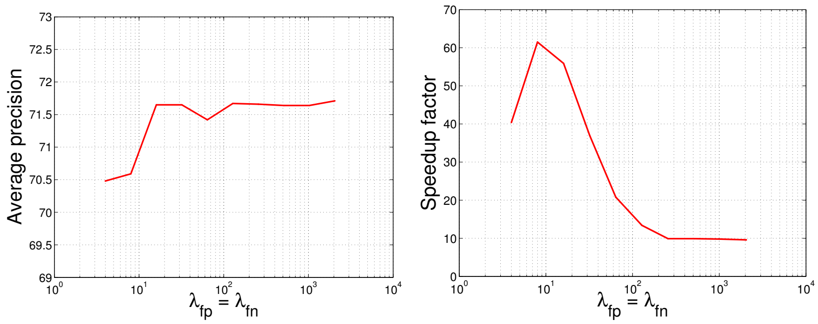


Figure 4.11: Average precision and relative number of part evaluations versus DPM as a function of the parameter $\lambda = \lambda_{fn} = \lambda_{fp}$ on a log scale. The curves are reported on the bus class from the VOC 2007 *training* set.

4.5.4 Comparison of DPM, Active DPM, and Cascade DPM

In this section we compare ADPM versus two baselines, DPM (Felzenszwalb et al. 2010b) and cascade DPM (Cascade, Felzenszwalb et al. (2010a)) in terms of average precision (AP), relative number of part evaluations (RNPE), and relative wall-clock time speedup (Speedup). Experiments were carried out on all 20 classes in the PASCAL VOC 2007 and 2010 datasets (Everingham et al. 2010). Publicly available PASCAL VOC 2007 and 2010 DPM and Cascade models were used for all three methods. For a fair comparison, ADPM changes only the part order and the stopping criterion of the original implementations.

ADPM vs DPM

The inference of ADPM on two input images is shown in detail in Fig. 4.10 and Fig. 4.12. The probability of a positive label $p_t(\oplus)$ (top row) becomes more contrasted as additional parts are evaluated. The number of locations at which the algorithm has not terminated decreases rapidly as time progresses. Visually, the locations with a maximal posterior are identical to the top scores obtained by the DPM. The order of parts chosen by the policy is indicative of their informativeness. For example, in Fig. 4.12 the wheel filters are applied first which agrees with intuition. In this example, the probability $p_t(\oplus)$ remains low at the correct location for several iterations due to the occlusions. Nevertheless, the policy recognizes that it should not terminate and, as it evaluates more parts, the correct location of the highest DPM score is reflected in the posterior. A video demonstrating the active DPM inference is provided in Appendix D.16 Extension 17.

ADPM was compared to DPM in terms of AP and RNPE to demonstrate the ability of ADPM to reduce the number of part evaluations with minimal loss in accuracy, irrespective of the features used. The parameters were set to $\lambda_{fp} = 20$ and $\lambda_{fn} = 5$ for all classes based on the analysis in Sec. 4.5.3. Table 4.4 shows that ADPM achieves a significant decrease (90 times on average) in the number of evaluated parts with negligible loss in accuracy. The precision-recall curves of the two methods are shown in Fig. 4.13 for several classes.

ADPM vs Cascade DPM

The improvement in detection speed achieved by ADPM is demonstrated via a comparison to Cascade in terms of AP, RNPE, and wall-clock time (in sec). During inference, Cascade

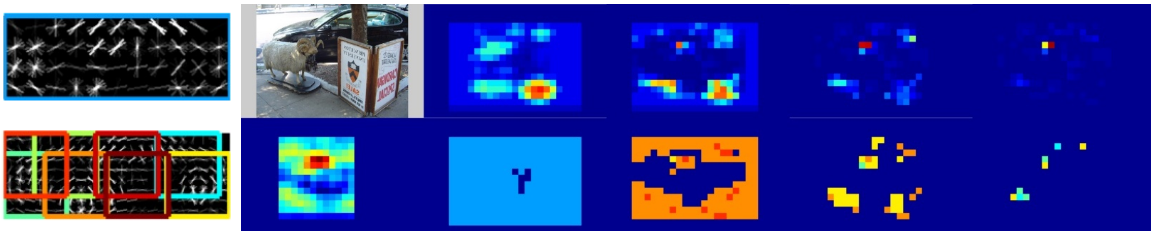
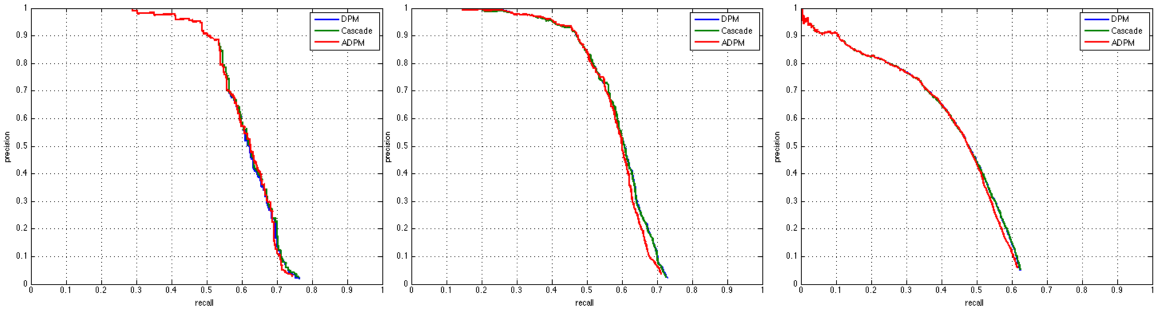


Figure 4.12: Illustration of the ADPM inference process on a car example. The DPM model with colored root and parts is shown on the left. The top row on the right consists of the input image and the evolution of the positive label probability $p_t(\oplus)$ for $t \in \{1, 2, 3, 4\}$ (high values are red; low values are blue). The bottom row consists of the full DPM $score(q)$ and a visualization of the parts applied at different locations at time t. The pixel colors correspond to the part colors on the left. In this example, despite the car being heavily occluded, ADPM converges to the correct location after four iterations. For more examples see the video in Appendix D.16 Extension 17.

Table 4.4: Average precision (AP) and relative number of part evaluations (RNPE) of DPM versus ADPM on all 20 classes in VOC 2007 and 2010.

VOC2007	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mean**
DPM RNPE	102.8	106.7	63.7	79.7	58.1	155.2	44.5	40.0	58.9	71.8	69.9	49.2	51.0	59.6	45.3	49.0	62.6	68.6	79.0	100.6	**70.8**
DPM AP	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	**33.7**
ADPM AP	33.5	59.8	9.8	15.3	27.6	52.5	57.6	22.1	20.1	24.6	24.9	12.3	57.6	48.4	42.8	12.0	20.4	35.7	46.3	43.2	**33.3**
VOC2010	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mean**
DPM RNPE	110.0	100.8	47.9	98.8	111.8	214.4	75.6	202.5	150.8	147.2	62.4	126.2	133.7	187.1	114.4	59.3	24.3	131.2	143.8	106.0	**117.4**
DPM AP	45.6	49.0	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.2	10.7	20.5	42.5	44.5	41.3	8.7	29.0	18.7	40.0	34.5	**29.6**
ADPM AP	45.3	49.1	10.2	12.2	26.9	50.6	41.9	22.7	16.5	22.8	10.6	19.7	40.8	44.5	36.8	8.3	29.1	18.6	39.7	34.5	**29.1**



(a) class: bicycle

(b) class: car

(c) class: person

Figure 4.13: Precision recall curves for bicycle, car, and person classes from VOC 2007. Our method's accuracy ties with the baselines.

prunes the image locations in two passes. In the first pass, the locations are filtered using the PCA filters and the low-scoring ones are discarded. In the second pass, the remaining locations are filtered using the full-dimensional filters. To make a fair comparison, we adopted a similar *two-stage* approach. An *additional* policy was learned using PCA score likelihoods and was used to schedule PCA filters during the first pass. The locations, which were selected as foreground in the first stage, were filtered again, using the original policy to schedule full-dimensional filters. The parameters λ_{fp} and λ_{fn} were set to 20 and 5 for the PCA policy and to 50 and 5 for the full-dimensional policy. A higher λ_{fp} was chosen to make the prediction more precise (albeit slower) during the second stage. Deformation pruning was not used for either method. Table 4.5 summarizes the results. A discrepancy in the speedup of ADPM versus Cascade is observed in Table 4.5. On average, ADPM is 7 times faster than Cascade in RNPE but only 3 times faster in seconds. A breakdown of the computational time during inference on a single image is shown in Table 4.6. We observe that the ratios of part evaluations and of seconds are consistent within individual stages (PCA and full). However, a single filter evaluation during the full-filter stage is significantly slower than one during the PCA stage. This does not affect the cumulative RNPE but lowers the combined seconds ratio. While ADPM is significantly faster than Cascade during the PCA stage, the speedup (in sec) is reduced during the slower full-dimensional stage.

Table 4.5: Average precision (AP), relative number of part evaluations (RNPE), and relative wall-clock time speedup (Speedup) of ADPM versus Cascade on all 20 classes in VOC 2007 and 2010.

VOC2007	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mean**
Cascade RNPE	5.93	5.35	9.17	6.09	8.14	3.06	5.61	4.51	6.30	4.03	4.83	7.77	3.61	6.67	17.8	9.84	3.82	2.43	2.89	6.97	**6.24**
ADPM Speedup	3.14	1.60	8.21	4.57	3.36	1.67	2.11	1.54	3.12	1.63	1.28	2.72	1.07	1.50	3.59	6.15	2.92	1.10	1.11	3.26	**2.78**
Cascade AP	33.2	60.8	10.2	16.1	27.3	54.1	58.1	23.0	20.0	24.2	26.8	12.7	58.1	48.2	43.2	12.0	20.1	35.8	46.0	43.4	**33.7**
ADPM AP	31.7	59.0	9.70	14.9	27.5	51.4	56.7	22.1	20.4	24.0	24.7	12.4	57.7	48.5	41.7	11.6	20.4	35.9	45.8	42.8	**33.0**
VOC2010	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mean**
Cascade RNPE	7.28	2.66	14.80	7.83	12.22	5.47	6.29	6.33	9.72	4.16	3.74	10.77	3.21	9.68	21.43	12.21	3.23	4.58	3.98	8.17	**7.89**
ADPM Speedup	2.15	1.28	7.58	5.93	4.68	2.79	2.28	2.44	3.72	2.42	1.52	2.76	1.57	2.93	4.72	8.24	1.42	1.81	1.47	3.41	**3.26**
Cascade AP	45.5	48.9	11.0	11.6	27.2	50.5	43.1	23.6	17.2	23.1	10.7	20.5	42.4	44.5	41.3	8.7	29.0	18.7	40.1	34.4	**29.6**
ADPM AP	44.5	49.2	9.5	11.6	25.9	50.6	41.7	22.5	16.9	22.0	9.8	19.8	41.1	45.1	40.2	7.4	28.5	18.3	38.0	34.5	**28.8**

Table 4.6: An example demonstrating the computational time breakdown during inference of ADPM and Cascade on a single image. The number of part evaluations (PE) and the inference time (in sec) is recorded for the PCA and the full-dimensional stages. The results are reported once without and once with cache use. The number of part evaluations is independent of caching.

	PCA no cache	PCA cache	PE	Full no cache	Full cache	PE	Total no cache	Total cache	Total PE
CASCADE	4.34s	0.67s	208K	0.13s	0.08s	1.1K	4.50s	0.79s	209K
ADPM	0.62s	0.06s	36K	0.06s	0.04s	0.6K	0.79s	0.19s	37K

4.6 Summary

Some information gathering applications such as object recognition necessitate discrete-space observation and motion models. Because these models are not “linearizable” the open-loop planning approach of Ch. 2 cannot be used. This chapter focused on nonmyopic closed-loop planning for discrete-space active information acquisition. It developed an exact algorithm, based on dynamic programming, and an approximation algorithm based on Monte Carlo tree search. Applications in camera view planning for object classification and pose estimation and in single-image object recognition via deformable part models were used to demonstrate the performance of the algorithms. The results showed that active approaches for camera view planning provides a significant improvement over static object recognition. Also, the main advantage of nonmyopic planning over greedy planning in this application turned out to be the adaptive stopping criterion that depends on the observations received online. In the single-image object detection application, the active part scheduling approach accelerated inference with pictorial structures substantially, without sacrificing accuracy. Unlike existing approaches, which use a pre-specified part order and hard stopping thresholds, our approach selects the part order and the stopping criterion adaptively, based on the filter responses obtained during inference.

In summary, the techniques developed in this chapter offer a nonmyopic closed-loop planning approach that may impact many robotics tasks that optimize information gathering in a discrete non-Gaussian setting.

Chapter 5

Conclusions and Future Work

Using autonomous sensor teams to collect information about physical phenomena of interest has the potential to revolutionize fields such as environmental monitoring, agriculture, construction and structure inspection, security and surveillance, search and rescue operations, localization and mapping, and mining. Effective coordination of the mobile sensing resources, aimed at improving the inference process, is one of the great technical challenges that inspired this dissertation. Due to the coupling between estimation and control in information gathering problems, there is a clear need for new planning and modeling techniques that can handle long planning horizons, distributed coordination among the sensing systems, and heterogeneity in the environment and sensor models. This dissertation offers a general formulation of the active information acquisition problem and develops scalable algorithms, accompanied by theoretical analysis, that are applicable to a broad variety of practical applications.

Ch. 2 argued that if the sensor observation models and the target motion models are "linearizable", the stochastic active information acquisition problem can be reduced to a deterministic optimal control problem, in which open-loop planning is optimal. Such a reduction is possible due to Thm. 2.1, which proves that a separation principle between estimation and control holds for the linear Gaussian active information acquisition problem. Still, the complexity of computing the optimal nonmyopic open-loop policy in the deterministic problem scales exponentially with the planning horizon T and the number of sensors n. The following contributions of Ch. 2 enable an efficient solution with performance guarantees:

- The complexity in the planning horizon T was mitigated by developing a reduced value iteration algorithm (Alg. 2), which approximates the optimal solution with finite-time suboptimality guarantees (Thm. 2.2, Thm. 2.3, Thm. 2.4).

- The complexity in the number of sensors n was reduced from exponential (in the centralized setting) to linear via decentralized control. The decentralized algorithm is guaranteed by Thm. 2.6 to obtain at least 50% of the information obtained by the optimal centralized solution.

- The target inference process was decentralized as well in order to complement the decentralized control scheme. We developed a distributed Kalman filter (Alg. 5) for dynamic target tracking and a distributed Jacobi algorithm (2.27) for sensor self-localization. It was shown in Thm. 2.12 that the two algorithms can be used in conjunction to achieve joint localization and estimation in sensor networks with arbitrarily small asymptotic mean square error.

- The linear Gaussian techniques above can even produce an adaptive policy for nonlinear motion and observation models via linearization and model predictive control (Sec. 2.4.6).
- Applications in methane emission monitoring, mobile vehicle tracking, distributed localization in sensor networks, and active multi-robot simultaneous localization and mapping were presented.

While the techniques of Ch. 2 capture a large class of information acquisition problems, they are not applicable in three particular situations: (1) when the data association between the received measurements and the observed targets is unknown, (2) when the sensor observation models are unknown, and (3) when the sensor and target states or the measurements are discrete.

These complications were considered by Ch. 3 in the context of two applications: active localization using object recognition as input and physical signal source seeking with unknown sensing models. We made the following contributions:

- A sensor observation model that incorporates missed detections, false alarms, and unknown data association was developed by considering set-valued observations.
- It was proven in Thm. 3.2 that the likelihood of a set-valued observation can be obtained via a matrix permanent computation. This allows a polynomial-time approximation to a Bayes filter with set-valued observations (Alg. 7 and Alg. 6).
- A stochastic gradient ascent algorithm was developed for multi-sensor information gathering with unknown observation and target motion models. We proved that a distributed finite-difference approximation of the gradient of the measured signal based on Gaussian radial bases functions is unbiased (Thm. 3.7). Moreover, the stochastic gradient ascent guarantees (Thm. 3.5) that the sensors converge to a small neighborhood around a local maximum of the measured signal.
- Applications in vehicle localization in residential areas using semantically-meaningful landmarks, hand-held device global localization, active robot localization using object recognition, and wireless radio source localization were presented.

Finally, Ch. 4 considered nonmyopic closed-loop planning in discrete state and measurement spaces and made the following contributions:

- Exact nonmyopic closed-loop planning for discrete-space information gathering was achieved via dynamic programming (Alg. 9). The algorithm obtains the optimal solution but scales poorly with the size of the state and measurement spaces.
- To provide scalability, an approximate algorithm based on Monte Carlo tree search with a rollout policy that exploits the structure of the information measure was developed (Alg. 10).

- Applications in object classification and pose estimation with a mobile depth camera (Alg. 11) and active object recognition with deformable part models (Alg. 12) were presented.

In summary, the techniques developed in this dissertation offer a scalable and versatile non-greedy solution to the general active information acquisition problem, accompanied by theoretical performance guarantees. Potential extensions to the current results that warrant further investigation in light of the long-term goals listed in Sec. 1.4 are summarized below.

- **Improved target inference:** Several new techniques for target inference were proposed in this dissertation. Among them were the particle filter with set-valued observations in Sec. 3.1.4, the distributed Kalman filter in Sec. 2.6.1, and the square root information filter in Sec. 2.4.5 that exploits sparsity in the target information matrix. An interesting direction for future research is to develop a unified target inference method that combines the benefits of the Kalman-type filters (no particle depletion problems in high dimensions; distributed computation; exploits sparsity) with those of the set-based particle filter (can handle both continuous and discrete measurements and unknown data association). Further, it would be advantageous to use ideas from graphical model inference (Koller and Friedman 2009, Sudderth et al. 2003, Meyer et al. 2014) to replace filtering with smoothing, which was shown to be superior in applications such as localization and mapping (Kaess 2008) because past linearization points and past target estimates can be corrected.

- **Anytime incremental informative planning:** This thesis developed efficient techniques for both open-loop (reduced value iteration, Sec. 2.4.4) and closed-loop (Monte Carlo tree search, Sec. 4.3) planning that can handle long planning horizons and provide performance guarantees. However, some aspects of these planners are still not at the maturity level of traditional planners. For example, future research can be focused on developing anytime (a valid solution can be provided at any time) incremental (reuses information from previous planning episodes) versions of the information-based planners, using inspiration from ARA* (Likhachev et al. 2003) and AD* (Likhachev et al. 2005).

- **Motion primitive generation:** The set of motion primitives available as control inputs for the sensors are an important aspect of the information planning process that received little attention in this dissertation. Future work should investigate how to design motion primitives suitable for a particular information gathering task or how to select a small subset of motion primitives from a potentially very large library of sensor controllers. It is also possible to generate or optimize the sensor motion primitives online (Kröger 2009). Finally, the informative planning algorithms should be extended to handle *multi-scale* motion primitives, which involve different numbers of sensing locations at different separations.

- **Experimental validation:** It is of great interest to carry out large-scale multi-robot experimental validation of the active information acquisition techniques and to explore other potential applications. Some obvious examples of the latter include active dense mapping (Newcombe 2012) and active semantic mapping but we believe that the techniques developed in this thesis can provide significant performance gains in many other applications and fields.

Appendix A

Information Measures

Entropy. *Let X be a discrete random variable with probability mass function p. Then, the entropy of X is defined as:*

$$\mathbf{H}(X) := -\sum_x p(x) \log p(x)$$

Differential Entropy. *Let X be a continuous random variable with probability density function p. Then, the entropy of X is defined as:*

$$\mathbf{h}(X) := -\int p(x) \log p(x) dx$$

Conditional Entropy. *Let X and Y be discrete random variables with joint probability mass function $p(x, y)$ and marginals $p(x)$ and $p(y)$, respectively. Then, the entropy of X conditioned on Y is defined as:*

$$\mathbf{H}(X \mid Y) := \sum_y p(y) \mathbf{H}(X \mid Y = y) = \sum_{x,y} p(x, y) \log \frac{p(x)}{p(x, y)}$$

Rényi Entropy. *Let X be a discrete random variable with probability mass function p. Then, the Rényi entropy of X of order α, where $\alpha \geq 0$ and $\alpha \neq 1$, is defined as:*

$$\mathbf{H}_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_x p(x)^\alpha \right)$$

Entropy Chain Rule (Cover and Thomas (2012))**.** *Let $X_1, \ldots, X_n$ be a collection of random variables. Then, their joint entropy satisfies:*

$$\begin{aligned}\mathbf{H}(X_1, \ldots, X_n) &= \mathbf{H}(X_n \mid X_1, \ldots, X_{n-1}) + \ldots + \mathbf{H}(X_2 \mid X_1) + \mathbf{H}(X_1) \\ &= \sum_{i=1}^n \mathbf{H}(X_i \mid X_1, \ldots, X_{i-1})\end{aligned}$$

Differential Entropy of a Gaussian Random Vector. *Let $X \sim \mathcal{N}(\mu, \Sigma)$ be n-dimensional. Then, its differential entropy is:*

$$\mathbf{h}(X) = \frac{1}{2} \left(n \log(2\pi e) + \log \det(\Sigma) \right)$$

Kullback-Leibler divergence. *Let p and q be probability density functions. Then, the Kullback-Leibler divergence of q from p is defined as:*

$$D(p\|q) := \int p(x) \log \frac{p(x)}{q(x)} dx$$

Mutual Information. *Let X and Y be discrete random variables with joint probability mass function $p(x, y)$ and marginals $p(x)$ and $p(y)$, respectively. Then, the mutual information of X and Y is defined as:*

$$\mathbf{I}(X;Y) := \sum_y \sum_x p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = \mathbf{H}(X) - \mathbf{H}(X \mid Y) = D(p(x,y)\|p(x)p(y))$$

Cauchy-Schwarz divergence. *Let p and q be probability density functions. Then, the Cauchy-Schwarz divergence of q from p is defined as:*

$$D_{CS}(p\|q) := -\log \left(\frac{\left(\int p(x)q(x)dx\right)^2}{\int p^2(x)dx \int q^2(x)dx} \right)$$

Cauchy-Schwarz Quadratic Mutual Information (Principe (2010))**.** *Let X and Y be discrete random variables with joint probability mass function $p(x, y)$ and marginals $p(x)$ and $p(y)$, respectively. Then, the Cauchy-Schwarz quadratic mutual information of X and Y is defined as:*

$$\mathbf{I}_{CS}(X;Y) := D_{CS}(p(x,y)\|p(x)p(y))$$

Mutual Information Chain Rule (Cover and Thomas (2012))**.** *Let X and $Y_1, \dots, Y_n$ be random variables. Then, the mutual information of X and $[Y_1, \dots, Y_n]$ satisfies:*

$$\mathbf{I}(X;Y_1,\dots,Y_n) = \sum_{i=1}^{n} \mathbf{I}(X;Y_i \mid Y_1,\dots,Y_{i-1})$$

Probability of Error. *Let $Z_1, \dots, Z_t$ be a sequence of random variables. Let Y be a discrete random variable with (random) conditional probability mass function $p_t(y) := \mathbb{P}(Y = y \mid Z_1, \dots, Z_t)$ Then, the probability of error of the maximum likelihood estimate $\hat{Y}_t := \arg\max_y p_t(y)$ is defined as:*

$$Pe(t) := \mathbb{P}(Y \neq \hat{Y}_t) = \mathbb{E}_{Z_{1:t}} \sum_y \left(\mathbb{1}_{\left\{y \neq \arg\max\limits_j p_t(j)\right\}} p_t(y) \right) = \mathbb{E}_{Z_{1:t}} \left(1 - \max_y p_t(y) \right)$$

Fano's Inequality (Cover and Thomas (2012))**.** *Let $Z_1, \dots, Z_t$ be a sequence of random variables. Let Y be a discrete random variable with support $\mathcal{Y}$. Then, the entropy of Y, conditioned on $Z_1, \dots, Z_t$, and the probability of error $Pe(t)$ of the maximum likelihood estimate of Y, given $Z_1, \dots, Z_t$, are related as follows:*

$$\mathbf{H}(Y \mid Z_{1:t}) \leq Pe(t) \log(|\mathcal{Y}| - 1) + \mathbf{H}(Pe(t))$$

Appendix B

Sensor and Motion Models

B.1 Differential-drive Motion Model

This section presents a discrete-time model of a differential-drive robot in the form $r_{t+1} = f(r_t, u_t)$, where $r_t := (x_t, y_t, \theta_t)$ is the pose of the robot at time t, consiting of its 2-D position and orientation, and u_t is the control input that is potentially perturbed by additive motion noise. Let ϵ be the time discretization step, l be the distance between the centers of the two wheels, and V_L, V_R be the left and right wheel velocities along the ground. The control input to the robot consits of the linear and angular velocities,

$$v_t = \frac{V_L + V_R}{2} \qquad \omega_t = \frac{V_R - V_L}{l}.$$

Further, we can define the instantaneous center of curvature (ICC), and the signed distance R_t from the ICC to the mid point between the wheels,

$$R_t = \frac{v_t}{w_t}, \qquad ICC = \begin{pmatrix} x_t - R_t \sin\theta_t & y_t + R_t \cos\theta_t \end{pmatrix}^T.$$

The discrete-time differential-drive model is:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{bmatrix} \cos(\epsilon\omega_t) & -\sin(\epsilon\omega_t) & 0 \\ \sin(\epsilon\omega_t) & \cos(\epsilon\omega_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_t - ICC_x \\ y_t - ICC_y \\ \theta_t \end{pmatrix} + \begin{pmatrix} ICC_x \\ ICC_y \\ \epsilon\omega_t \end{pmatrix}.$$

Note that as $\epsilon\omega_t \to 0$, $R_t \to \infty$, and we can simplify the above:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} + \begin{cases} \begin{pmatrix} R_t(\sin(\theta_t + \epsilon\omega_t) - \sin\theta_t) \\ R_t(\cos\theta_t - \cos(\theta_t + \epsilon\omega_t)) \\ \epsilon\omega_t \end{pmatrix}, & |\epsilon\omega_t| \geq \delta \\ \begin{pmatrix} \epsilon v_t \cos(\theta_t + \epsilon\omega_t/2) \\ \epsilon v_t \sin(\theta_t + \epsilon\omega_t/2) \\ \epsilon\omega_t \end{pmatrix}, & |\epsilon\omega_t| < \delta \end{cases}.$$

where δ is a small constant. The trajecotries of the differential-drive motion model can be parametrized by distance $s_t := \epsilon v_t$ and curvature $\kappa_t := \frac{1}{R_t} = \frac{\omega_t}{v_t}$ as well.

To use the above model for some estimation and control tasks, it is necessary to linearize the equations. The Jacobian of the motion model with respect to (x_t, y_t, θ_t) is:

$$\begin{cases} \begin{bmatrix} 1 & 0 & R_t(\cos(\theta_t + \epsilon\omega_t) - \cos\theta_t) \\ 0 & 1 & R_t(-\sin\theta_t + \sin(\theta_t + \epsilon\omega_t)) \\ 0 & 0 & 1 \end{bmatrix}, & |\epsilon\omega_t| \geq \delta \\ \begin{bmatrix} 1 & 0 & -\epsilon v_t \sin(\theta_t + \epsilon\omega_t/2) \\ 0 & 1 & \epsilon v_t \cos(\theta_t + \epsilon\omega_t/2) \\ 0 & 0 & 1 \end{bmatrix}, & |\epsilon\omega_t| < \delta \end{cases},$$

while with respect to $u_t := [v_t\ \omega_t]^T$ it is:

$$\begin{cases} \begin{bmatrix} \frac{1}{\omega_t}(\sin(\theta_t + \epsilon\omega_t) - \sin\theta_t) & \frac{-v_t \sin(\theta_t+\epsilon\omega_t)+v_t \sin\theta_t}{\omega_t^2} + \frac{\epsilon v_t \cos(\theta_t+\epsilon\omega_t)}{\omega_t} \\ \frac{1}{\omega_t}(\cos\theta_t - \cos(\theta_t + \epsilon\omega_t)) & \frac{-v_t \cos\theta_t+v_t \cos(\theta_t+\epsilon\omega_t)}{\omega_t^2} + \frac{\epsilon v_t \sin(\theta_t+\epsilon\omega_t)}{\omega_t} \\ 0 & \epsilon \end{bmatrix}, & |\epsilon\omega_t| \geq \delta \\ \begin{bmatrix} \epsilon\cos(\theta_t + \epsilon\omega_t/2) & -\frac{\epsilon^2}{2} v_t \sin(\theta_t + \epsilon\omega_t/2) \\ \epsilon\sin(\theta_t + \epsilon\omega_t/2) & \frac{\epsilon^2}{2} v_t \cos(\theta_t + \epsilon\omega_t/2) \\ 0 & \epsilon \end{bmatrix}, & |\epsilon\omega_t| < \delta \end{cases}.$$

B.2 Double-integrator Motion Model

Consider a vehicle with position $x(t) \in \mathbb{R}^2$ and velocity $v(t) \in \mathbb{R}^2$. If the velocity can be controlled, we obtain a double-integrator motion model in continuous time:

$$\dot{x} = v, \;\; \dot{v} = u(t) + w(t),$$

where $u(t)$ is the control input and $w(t)$ is a zero-mean white Gaussian process with power spectral density qI_2. Hence, $\mathbb{E}w(t) = 0$ and the autocorrelation of the noise process is $\mathbb{E}[w(t)w(\tau)^T] = qI_2\delta(t - \tau)$, where δ is the Dirac delta function, and q is a scalar diffusion strength measured in $(m/sec^2)^2 \frac{1}{Hz}$. In state-space form, the equations look like:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{v}(t) \end{pmatrix} = \begin{bmatrix} 0 & I_2 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} + \begin{bmatrix} 0 \\ I_2 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ I_2 \end{bmatrix} w(t)$$

Denote the first and the second matrices above by $A \in \mathbb{R}^{4\times4}$ and $B \in \mathbb{R}^{4\times2}$, respectively.

The system can be discretized with sampling period ϵ as follows:

$$\begin{aligned} \begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} &= e^{\epsilon A} \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \left[\int_0^\epsilon e^{(\epsilon-s)A} B ds\right] u_k + \mathcal{N}\left(0, q\int_0^\epsilon e^{(\epsilon-s)A} BB^T e^{(\epsilon-s)A^T} ds\right) \\ &= \begin{bmatrix} I_2 & \epsilon I_2 \\ 0 & I_2 \end{bmatrix} \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \begin{bmatrix} \epsilon I_2 & \epsilon^2/2I_2 \\ 0 & \epsilon I_2 \end{bmatrix} u_k + \mathcal{N}\left(0, q \begin{bmatrix} \epsilon^3/3I_2 & \epsilon^2/2I_2 \\ \epsilon^2/2I_2 & \epsilon I_2 \end{bmatrix}\right) \end{aligned}$$

where it is assumed that between samples the input is kept constant so that $u_k = u(\epsilon t)$, $x_k = x(\epsilon t)$, and $v_k = v(\epsilon t)$. We can of course take this a step further and define a constant-acceleration motion model:

$$\dot{x} = v, \;\; \dot{v} = a(t), \;\; \dot{a} = u(t) + w(t),$$

and discretize it as before.

When modeling the motion of a target in tracking applications, the control input $u(t)$ is unknown and the second term in the discrete-time motion model should be omitted.

B.3 Range Sensor Observation Model

This section models the operation of a sensor with position $x \in \mathbb{R}^d$ which measures the distance to a landmark with position $y \in \mathbb{R}^d$. The model has the form:

$$z = h(x, y) + v := \|x - y\|_2 + v,$$

where $z \in \mathbb{R}_{\geq 0}$ is the measurement and v is the measurement noise. In some applications (e.g., the linear Gaussian framework in Ch. 2), it is necessary to use a linearized model. The Jacobians are:

$$\nabla_x h(x, y) = \frac{1}{h(x, y)}(x - y)^T \qquad \nabla_y h(x, y) = \frac{1}{h(x, y)}(y - x)^T.$$

B.4 Bearing Sensor Observation Model

This section models the operation of a sensor with pose $x := [x_1\ x_2\ \theta]^T \in SE(2)$ which measures the bearing (in the body frame) to a landmark with position $y := [y_1\ y_2]^T \in \mathbb{R}^2$. The model has the form:

$$z = h(x, y) + v := \arctan\left(\frac{y_2 - x_2}{y_1 - x_1}\right) - \theta + v,$$

where $z \in SO(2)$ is the measurement and v is the measurement noise. In some applications (e.g., the linear Gaussian framework in Ch. 2), it is necessary to use a linearized model. The Jacobians are:

$$\nabla_x h(x, y) = \begin{bmatrix} \frac{y_2 - x_2}{\|y - x\|_2^2} & \frac{x_1 - y_1}{\|y - x\|_2^2} & -1 \end{bmatrix} \qquad \nabla_y h(x, y) = \frac{1}{\|x - y\|_2^2} \begin{bmatrix} (x_2 - y_2) & (y_1 - x_1). \end{bmatrix}$$

B.5 Stereo Sensor Observation Model

This section models the operation of a planar stereo sensor with pose $x := [x_1\ x_2\ \theta]^T \in SE(2)$, baseline b, and focal length f. Consider a landmark with position $y := [y_1\ y_2]^T \in \mathbb{R}^2$ in the global frame. The sensor measures the landmark position in the left and right image lines (instead of planes since the sensor is planar) according to the model:

$$\begin{aligned} z = h(x, y) + v &= \frac{f}{2p_z(x, y)} \begin{bmatrix} 2p_x(x, y) + b \\ 2p_x(x, y) - b \end{bmatrix} + v \\ p_z(x, y) &:= (\cos\theta)(y_2 - x_2) - (\sin\theta)(y_1 - x_1) \\ p_x(x, y) &:= (\cos\theta)(y_1 - x_1) - (\sin\theta)(y_2 - x_2) \end{aligned}$$

where $z \in \mathbb{R}^2$ is the measurement of the left and right point images, respectively, and v is the measurement noise. In some applications (e.g., the linear Gaussian framework in Ch. 2), it is necessary to use a linearized model. The Jacobians are:

$$
\begin{aligned}
\nabla_x h(x,y) &= \begin{bmatrix} \frac{\partial h_1(x,y)}{\partial x_1} & \frac{\partial h_1(x,y)}{\partial x_2} & \frac{\partial h_1(x,y)}{\partial \theta} \\ \frac{\partial h_2(x,y)}{\partial x_1} & \frac{\partial h_2(x,y)}{\partial x_2} & \frac{\partial h_1(x,y)}{\partial \theta} \end{bmatrix} \\
\frac{\partial h_1(x,y)}{\partial x_1} &= -f\Bigg(\frac{\cos\theta}{p_z(x,y)} + \frac{\sin\theta(2p_x(x,y)+b)}{2p_z^2(x,y)}\Bigg) \\
\frac{\partial h_1(x,y)}{\partial x_2} &= f\Bigg(\frac{\sin\theta}{p_z(x,y)} + \frac{\cos\theta(2p_x(x,y)+b)}{2p_z^2(x,y)}\Bigg) \\
\frac{\partial h_1(x,y)}{\partial \theta} &= f\Bigg(\frac{(\sin\theta)(x_1-y_1)+(\cos\theta)(x_2-y_2)}{p_z(x,y)} \\
&\quad - \frac{((\cos\theta)(x_1-y_1)+(\sin\theta)(x_2-y_2))\,(2p_x(x,y)+b)}{2p_z^2(x,y)}\Bigg) \\
\frac{\partial h_2(x,y)}{\partial x_1} &= -f\Bigg(\frac{\cos\theta}{p_z(x,y)} + \frac{\sin\theta(2p_x(x,y)-b)}{2p_z^2(x,y)}\Bigg) \\
\frac{\partial h_2(x,y)}{\partial x_2} &= f\Bigg(\frac{\sin\theta}{p_z(x,y)} + \frac{\cos\theta(2p_x(x,y)-b)}{2p_z^2(x,y)}\Bigg) \\
\frac{\partial h_2(x,y)}{\partial \theta} &= f\Bigg(\frac{(\sin\theta)(x_1-y_1)+(\cos\theta)(x_2-y_2)}{p_z(x,y)} \\
&\quad - \frac{((\cos\theta)(x_1-y_1)+(\sin\theta)(x_2-y_2))\,(2p_x(x,y)-b)}{2p_z^2(x,y)}\Bigg) \\
\nabla_y h(x,y) &= f\begin{bmatrix} \left(\frac{\cos\theta}{p_z(x,y)} + \frac{\sin\theta(2p_x(x,y)+b)}{2p_z^2(x,y)}\right) & \left(\frac{-\sin\theta}{p_z(x,y)} - \frac{\cos\theta(2p_x(x,y)+b)}{2p_z^2(x,y)}\right) \\ \left(\frac{\cos\theta}{p_z(x,y)} + \frac{\sin\theta(2p_x(x,y)-b)}{2p_z^2(x,y)}\right) & \left(\frac{-\sin\theta}{p_z(x,y)} - \frac{\cos\theta(2p_x(x,y)-b)}{2p_z^2(x,y)}\right) \end{bmatrix}
\end{aligned}
$$

B.6 Relative-pose Sensor Observation Model

2-D case

Let $x := (p,\theta)$ be an element of $SE(2)$ with position $p \in \mathbb{R}^2$ and orientation $\theta \in SO(2)$. The model of a sensor which measures the relative transformation between two poses $x_t, x_{t+1} \in SE(2)$ is:

$$
h(x_{t+1}, x_t) := x_{t+1} \ominus x_t := \begin{bmatrix} R^T(\theta_t)(p_{t+1}-p_t) \\ \theta_{t+1}-\theta_t \end{bmatrix},
$$

where $R(\theta_t)$ is the rotation matrix associated with θ_t. The Jacobians are:

$$
\frac{\partial h}{\partial x_t} = \begin{bmatrix} -R^T(\theta_t) & \left[\frac{dR^T(\theta)}{d\theta}|_{\theta=\theta_t}\right](p_{t+1}-p_t) \\ 0_{1\times 2} & -1 \end{bmatrix} \qquad \frac{\partial h}{\partial x_{t+1}} = \begin{bmatrix} R^T(\theta_t) & 0_{2\times 1} \\ 0_{1\times 2} & 1 \end{bmatrix}.
$$

3-D case

Let $x := (p,q)$ be an element of $SE(3)$ with position $p \in \mathbb{R}^3$ and quaternion-represented orientation q with $\|q\| = 1$. The model of a sensor which measures the relative transformation between two poses $x_t, x_{t+1} \in SE(3)$ is:

$$h(x_{t+1}, x_t) := x_{t+1} \ominus x_t := \begin{bmatrix} R^T(q_t)(p_{t+1} - p_t) \\ q_t^{-1} q_{t+1} \end{bmatrix}$$

where $R^T(q_t)$ is the rotation matrix associated with q_t. The Jacobians with respect to position are:

$$\frac{\partial h}{\partial p_t} = \begin{bmatrix} -R^T(q_t) \\ 0_{3\times 3} \end{bmatrix} \qquad \frac{\partial h}{\partial p_{t+1}} = \begin{bmatrix} R^T(q_t) \\ 0_{3\times 3} \end{bmatrix}.$$

To obtain the jacobians with respect to orientation, note that quaternion multiplication can also be written in terms of matrix multiplications (Trawny and Roumeliotis 2005) as follows $q_{t+1} q_t = \mathcal{L}(q_{t+1}) q_t = \mathcal{R}(q_t) q_{t+1}$, where:

$$\mathcal{L}(q) := \begin{bmatrix} sI - \hat{v} & v \\ -v^T & s \end{bmatrix} \qquad \mathcal{R}(q) := \begin{bmatrix} sI + \hat{v} & v \\ -v^T & s \end{bmatrix},$$

and v is the vector part, $s := \sqrt{1 - \|v\|^2}$ is the scalar part of the unit quaternion $q := [v \; s]$, and $\hat{v}$ is the skew-symmetric matrix associated with v. Letting $q_{t+1} := [v_{t+1} \;\; \sqrt{1 - \|v_{t+1}\|^2}]$, we have:

$$\frac{\partial h}{\partial v_{t+1}} = \begin{bmatrix} 0_{3\times 3} \\ \mathcal{L}(q_t^{-1}) \frac{\partial}{\partial v_{t+1}} \begin{pmatrix} v_{t+1} \\ s_{t+1} \end{pmatrix} \end{bmatrix} = \begin{bmatrix} 0_{3\times 3} \\ \mathcal{L}(q_t^{-1}) \begin{pmatrix} I_{3\times 3} \\ -v_{t+1}^T / s_{t+1} \end{pmatrix} \end{bmatrix}$$

Similarly,

$$\frac{\partial q_t^{-1} q_{t+1}}{\partial v_t} = \mathcal{R}(q_{t+1}) \frac{\partial}{\partial v_t} \begin{pmatrix} -v_t \\ s_t \end{pmatrix} = \mathcal{R}(q_{t+1}) \begin{pmatrix} -I_{3\times 3} \\ -v_t^T / s_t \end{pmatrix}$$

Finally, for the translation component of h, note that $R^T(q_t) = (2s_t^2 - 1) I_{3\times 3} + 2 s_t \hat{v}_t + 2 v_t v_t^T$ (Trawny and Roumeliotis 2005) so that:

$$\begin{aligned} \frac{\partial R^T(q_t)(p_{t+1} - p_t)}{\partial v_t} &= -4 v_t^T (p_{t+1} - p_t) - 2 \frac{\partial}{\partial v_t} \left[s_t \widehat{(p_{t+1} - p_t)} v_t \right] + 4 v_t^T (p_{t+1} - p_t) \\ &= \frac{2}{s_t} \left(v_t^T \widehat{(p_{t+1} - p_t)} v_t - s_t^2 \widehat{(p_{t+1} - p_t)} \right) \end{aligned}$$

Appendix C

Bayesian Filtering

Recursive Bayesian estimation is a probabilistic approach for estimating the probability density (or mass) function (pdf) of a random variable over time, using incoming measurements, an observation model, and a motion model. To be more precise, given motion and observation models:

$$y_{t+1} = a(y_t, \text{noise}), \qquad z_t = h(y_t, \text{noise}),$$

the evolution of the variable of interest, y_t, is modeled via a hidden Markov model (Koller and Friedman 2009). The transition probabilities $p_a(y_{t+1} \mid y_t)$ are derived from the motion model, while the emission probabilities $p_h(z_t \mid y_t)$ - from the observation model. Using the Markov assumptions (that current state given the immediately previous one is conditionally independent of the earlier states and that the current measurement is dependent only upon the current state), the joint pdf of all states and measurements decomposes as:

$$p(y_{0:t}, z_{0:t}) = p_{0|0}(y_0) \prod_{k=0}^{t} p_h(z_k \mid y_k) \prod_{k=1}^{t} p_a(y_k \mid y_{k-1}),$$

where $p_{0|0}$ is a prior pdf. For any t, denote the pdf of y_t conditioned on the past measurements $z_{0:t}$ by $p_{t|t}$ and that of $y_{t+1} \mid z_{0:t}$ by $p_{t+1|t}$. The goal of the Bayes filter is to track the evolution of $p_{t|t}$, which reduces to iterating the following two steps:

$$\textbf{\textit{Predict:}} \qquad p_{t+1|t}(y) = \int p_a(y \mid s) p_{t|t}(s) ds \tag{C.1}$$

$$\textbf{\textit{Update:}} \qquad p_{t+1|t+1}(y) = \frac{p_h(z_{t+1} \mid y) p_{t+1|t}(y)}{\int p_h(z_{t+1} \mid s) p_{t+1|t}(s) ds} \tag{C.2}$$

While the Bayesian recursion is theoretically appealing, its implementation requires a tractable representation of the pdfs $p_{t|t}, p_{t+1|t}$ and a tractable way to carry out the integrations above, which inevitably leads to approximations (Huber 2015). The sections below review two of the most widely-used approximations to the Bayes filter.

C.1 Kalman Filter

The Kalman filter uses a parametric distribution, namely Gaussian, to represent the pdf $p_{t|t}$ tractably, which also conveniently leads to a closed-form expression for the integrals in the prediction (C.1) and update (C.2) steps of the Bayes filter above. The assumption of Gaussian pdfs solves both difficulties associated with the Bayes filter. The price paid in exchange is that the true pdfs are actually Gaussian only for linear motion and observation models perturbed by Gaussian noise. Hence, whenever the models deviate from the linear Gaussian assumptions, the Kalman filter is merely an approximation to the Bayes filter.

First, we review the linear Gaussian case in which the Kalman filter and the Bayes filter are equivalent. Suppose that the motion model of y_t is linear

$$y_{t+1} = A_t y_t + w_t$$

and corrupted by additive Gaussian process noise $w_t \sim \mathcal{N}(0, W_t)$. Similarly, assume that the observation model is:

$$z_y = H_t y_t + v_t, \quad v_t \sim \mathcal{N}(0, V_t).$$

Finally, let the prior pdf $p_{0|0}$ be a Gaussian pdf, denoted $\phi(\,\cdot\,; \mu_{0|0}, \Sigma_{0|0})$, with mean $\mu_{0|0}$ and covariance $\Sigma_{0|0}$. Using these assumptions, the prediction (C.1) and update steps (C.2) of the Bayes filter can be evaluated in closed-form and interestingly $p_{t|t}$ remains Gaussian:

$$\begin{aligned}
\textbf{\textit{Predict:}} \qquad p_{t+1|t}(y) &= \int \phi(y; A_t s, W_t)\phi(s; \mu_{t|t}, \Sigma_{t|t}) ds \\
&= \phi(y; A_t \mu_{t|t}, A_t \Sigma_{t|t} A_t^T + W_t) \\
&=: \phi(y; \mu_{t+1|t}, \Sigma_{t+1|t}) \\
\textbf{\textit{Update:}} \quad p_{t+1|t+1}(y) &= \frac{\phi(z_{t+1}; H_{t+1} y, V_{t+1})\phi(y; \mu_{t+1|t}, \Sigma_{t+1|t})}{\int \phi(z_{t+1}; H_{t+1} s, V_{t+1})\phi(s; \mu_{t+1|t}, \Sigma_{t+1|t}) ds} \\
&= \phi\left(y; \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - H_{t+1}\mu_{t+1|t}), (I - K_{t+1|t}H_{t+1})\Sigma_{t+1|t}\right)
\end{aligned}$$

where $K_{t+1|t} := \Sigma_{t+1|t} H_{t+1}^T (H_{t+1}\Sigma_{t+1|t} H_{t+1}^T + V_{t+1})^{-1}$ is the Kalman gain.

C.2 Extended Kalman Filter

As mentioned earlier, the Kalman filter is an exact implementation of the Bayes filter only for linear Gaussian models. The extended Kalman filter (EKF) exploits linearization to apply the Kalman filter idea to differentiable nonlinear models perturbed by zero-mean continuous (non-Gaussian) noise with an existing second moment. The EKF is only an approximation to the Bayes filter. To be more precise, suppose that the models are now nonlinear:

$$\begin{aligned}
y_{t+1} &= a_t(y_t, w_t), \quad \mathbb{E}w_t = 0, \quad \mathbb{E}[w_t w_s^T] = W_t \delta(t-s) \\
z_t &= h_t(y_t, v_t), \quad \mathbb{E}v_t = 0, \quad \mathbb{E}[v_t v_s^T] = V_t \delta(t-s)
\end{aligned}$$

and define the Jacobians:

$$\begin{aligned}
A_t(\hat{y}) &:= \frac{\partial a_t(y,0)}{\partial y}\Big|_{y=\hat{y}} \qquad & B_t(\hat{y}) &:= \frac{\partial a_t(y,w)}{\partial w}\Big|_{y=\hat{y},\, w=0} \\
H_t(\hat{y}) &:= \frac{\partial h_t(y,0)}{\partial y}\Big|_{y=\hat{y}} \qquad & G_t(\hat{y}) &:= \frac{\partial h_t(y,v)}{\partial v}\Big|_{y=\hat{y},\, v=0}
\end{aligned}$$

The EKF forces the pdf $p_{t|t}$ to be Gaussian by using moment matching. In other words, it replaces the true $p_{t|t}$ with a Gaussian pdf with mean and covariance corresponding to those of $p_{t|t}$. The EKF equations are given below:

$$\begin{aligned}
\textbf{\textit{Predict:}} \qquad p_{t+1|t}(y) &\approx \int \phi(y; a_t(s,0), B_t(s)W_tB_t^T(s))\phi(s; \mu_{t|t}, \Sigma_{t|t})ds \\
&\approx \phi\left(y; a_t(\mu_{t|t},0), A_t(\mu_{t|t})\Sigma_{t|t}A_t^T(\mu_{t|t}) + B_t(\mu_{t|t})W_tB_t^T(\mu_{t|t})\right) \\
&=: \phi(y; \mu_{t+1|t}, \Sigma_{t+1|t}) \\
\textbf{\textit{Update:}} \quad p_{t+1|t+1}(y) &\approx \frac{\phi(z_{t+1}; h_{t+1}(y,0), G_{t+1}(y)V_{t+1}G_{t+1}^T(y))\phi(y; \mu_{t+1|t}, \Sigma_{t+1|t})}{\int \phi(z_{t+1}; h_{t+1}(s,0), G_{t+1}(s)V_{t+1}G_{t+1}^T(s))\phi(s; \mu_{t+1|t}, \Sigma_{t+1|t})ds} \\
&\approx \phi\left(y; \mu_{t+1|t} + K_{t+1|t}(z_{t+1} - h_{t+1}(\mu_{t+1|t},0)), (I - K_{t+1|t}H_{t+1}(\mu_{t+1|t}))\Sigma_{t+1|t}\right)
\end{aligned}$$

where the Kalman gain $K_{t+1|t}$ is defined as:

$$\Sigma_{t+1|t}H_{t+1}^T(\mu_{t+1|t})(H_{t+1}(\mu_{t+1|t})\Sigma_{t+1|t}H_{t+1}^T(\mu_{t+1|t}) + G_{t+1}(\mu_{t+1|t})V_{t+1}G_{t+1}^T(\mu_{t+1|t}))^{-1}.$$

Instead of direct linearization, another way of applying the Kalman filter to nonlinear models is to represent the distribution of y_t via weighted samples, called *sigma points*, and propagate them through the nonlinear equations. Filters, such as the unscented Kalman filter, based on this idea are called linear regression Kalman filters (Huber 2015). They do not require differentiability of the models and often result in more accurate approximations.

C.3 Particle Filter

Sometimes the nonlinearities in the motion and observation models are large enough to introduce multiple modes in the pdf $p_{t|t}$. In this case, approximating $p_{t|t}$ via a Gaussian distribution would not be accurate. Another approximation to the Bayes filter which can handle multimodal distributions is the particle filter. The particle filter represents the pdf $p_{t|t}$ via a set of weighted samples (particles) $\{y_{t|t}^i, w_{t|t}^i\}_{i=1}^{N_{t|t}}$ as follows:

$$p_{t|t}(y) = \sum_{i=1}^{N_{t|t}} w_{t|t}^i \delta(y - y_{t|t}^i)$$

where δ is the Dirac delta function and $\sum_{i=1}^{N_{t|t}} w_{t|t}^i = 1$. The mean and covariance of this distribution are

$$\mu_{t|t} := \sum_{i=1}^{N_{t|t}} w_{t|t}^i y_{t|t}^i \quad \text{and} \quad \Sigma_{t|t} := \sum_{i=1}^{N_{t|t}} w_{t|t}^i y_{t|t}^i (y_{t|t}^i)^T - \mu_{t|t}\mu_{t|t}^T,$$

respectively. As in the Kalman filter case, this has the additional benefit of simplifying the integrations in the prediction (C.1) and update (C.2) steps. The prediction step is:

$$\textbf{\textit{Predict:}} \qquad p_{t+1|t}(y) = \int p_a(y \mid s) \sum_{i=1}^{N_{t|t}} w_{t|t}^i \delta(s - y_{t|t}^i)ds = \sum_{i=1}^{N_{t|t}} w_{t|t}^i p_a(y \mid y_{t|t}^i)$$

The pdf $p_{t+1|t}$ turns out to be a mixture model with $N_{t|t}$ components. To keep a finite-dimensional representation, the pdf can be approximated with a weighted particle set again as follows:

1. Draw a sample $j \in \{1, \ldots, N_{t|t}\}$ from the probability mass function $[w^1_{t|t} \ \cdots \ w^{N_{t|t}}_{t|t}]$
2. Draw a sample $y_{t+1|t}$ from the motion model $p_a(\cdot \mid y^j_{t|t})$
3. Repeat $N_{t+1|t}$ to form the weighted particle set $\{y^i_{t+1|t}, w^i_{t+1|t}\}$ with $w^i_{t+1|t} := p_{t+1|t}(y^i_{t+1|t})$ and $i = 1, \ldots, N_{t+1|t}$, where $N_{t+1|t}$ is a free parameter. Commonly, $N_{t+1|t} = N_{t|t}$ and instead of sampling in step 1, each particle index is chosen once.

After the above procedure, we have the following approximation:

$$p_{t+1|t}(y) \approx \sum_{i=1}^{N_{t+1|t}} w^i_{t+1|t}\delta(y - y^i_{t+1|t})$$

and can proceed with the update step:

$$\textbf{\textit{Update:}} \qquad p_{t+1|t+1}(y) = \frac{p_h(z_{t+1} \mid y) \sum_{i=1}^{N_{t+1|t}} w^i_{t+1|t}\delta(y - y^i_{t+1|t})}{\int p_h(z_{t+1} \mid s) \sum_{j=1}^{N_{t+1|t}} w^j_{t+1|t}\delta(s - y^j_{t+1|t})ds}$$

$$= \sum_{i=1}^{N_{t+1|t}} \left[\frac{p_h(z_{t+1} \mid y^i_{t+1|t})w^i_{t+1|t}}{\sum_{j=1}^{N_{t+1|t}} p_h(z_{t+1} \mid y^j_{t+1|t})w^j_{t+1|t}} \right] \delta(y - y^i_{t+1|t})$$

which keeps the posterior finite-dimensional. There are, of course, different ways to approximate $p_{t+1|t}$ and $p_{t+1|t+1}$, which gives rise to different particle filter variants. The approximation procedure above uses the motion model $p_a(\cdot \mid y^i_{t|t})$ as a proposal distribution to draw particles and approximate the pdf $p_{t+1|t}$. This variant is known as a bootstrap filter (Gordon et al. 1993, Thrun et al. 2005).

Regardless of the employed proposal distribution, every particle filter uses *resampling* to avoid particle depletion - a situation in which most of the particle weights are close to zero. A resampling procedure is applied at time t if the effective number of particles $N_{eff} := 1/\sum_{i=1}^{N_{t|t}} \left(w^i_{t|t}\right)^2$ is less than a given threshold. Then, $N_{t|t}$ particles are drawn with replacement from the particle set $\{w^i_{t|t}, y^i_{t|t}\}$ with probability proportional to the weights. The weights of the new particles are set to $1/N_{t|t}$. Resampling methods differ in the way the new particles are drawn from the set $\{w^i_{t|t}, y^i_{t|t}\}$. Stratified resampling is optimal in terms of variance (Thrun et al. 2005). Instead of selecting samples independently from each other in the resampling process, the selection involves a sequential stochastic process summarized in the algorithm below.

Algorithm 13 Low Variance Sampler($\{y^i, w^i\}_{i=1}^N$)

```
$r = rand(0; 1/N)$,    $c = w^1$,    $k = 1$
for $i = 1 : N$ do
    $U = r + (i-1)/N$
    while $U > c$ do
        $k = k + 1$,    $c = c + w^k$
    Add $y^k$ to the new set
```

Appendix D

Proofs and Supplementary Material

D.1 Proof of Theorem 2.1

At any time t, the distribution of the target state y_t conditioned on any fixed measurement sequence $\hat{z}_{1:t}$ and control sequence $\hat{u}_{1:t}$ can be obtained via the Bayes filter (Appendix C). Due to the assumptions in (2.1) that y_0 is Gaussian, the target motion model is linear in y_t and Gaussian, and the sensor observation model is linear in y_t and Gaussian, the Bayes filter can be implemented exactly via the Kalman filter. In particular, the distribution of $y_t \mid \hat{z}_{1:t}, \hat{u}_{1:t}$ remains Gaussian for any $t \geq 0$ and its covariance Σ_t is independent of $\hat{z}_{1:t}$ and can be obtained by the Riccati map, given in (2.4) and (2.5). The differential entropy of an n-dimensional Gaussian random vector $X \sim \mathcal{N}(\mu, \Sigma)$ is $\mathbf{h}(X) = \frac{1}{2}\left(n \log(2\pi e) + \log\det(\Sigma)\right)$. Since the target distribution is Gaussian, the differential entropy term in the cost function in (2.1) satisfies for any t:

$$\begin{aligned}
\mathbf{h}(y_t \mid z_{1:t}) &= \mathbb{E}_{\hat{z}_{1:t}} \mathbf{h}(y_t \mid z_{1:t} = \hat{z}_{1:t}) \\
&= \frac{1}{2} \mathbb{E}_{\hat{z}_{1:t}} \Big(\log(2\pi e)^{d_y} + \log\det \Sigma_t(\hat{u}_{1:t}) \Big) \\
&= \frac{1}{2} \Big(\log(2\pi e)^{d_y} + \log\det \Sigma_t(\hat{u}_{1:t}) \Big) ,
\end{aligned} \tag{D.1}$$

where d_y is the dimension of y_t and we have emphasized that Σ_t depends on the control sequence $\hat{u}_{1:t}$ but not on the measurement realization $\hat{z}_{1:t}$. Let $\mu^* = \{\mu_0^*, .., \mu_{T-1}^*\}$ be optimal control policy in (2.1) with associated cost J^*. Fix a realization $\hat{z}_{1:T}$ of the measurements and let σ be the open-loop policy induced by μ^* given $\hat{z}_{1:T}$ with associated cost J^σ. From (D.1), J^* is independent of $\hat{z}_{1:T}$, hence $J^* = J^\sigma$ for any $\hat{z}_{1:T}$ and σ is optimal. □

D.2 Proof of Theorem 2.2

Definition D.1 (t-step Riccati map)**.** Given a control sequence $\sigma \in U^T$, the corresponding sensor state trajectory $\pi := x_0, \ldots, x_T \in \mathcal{X}^{T+1}$, and a covariance matrix $\Sigma \succeq 0$, define the t-step Riccati map recursively as follows:

$$\phi^0_{\sigma,\pi}(\Sigma) := \Sigma \qquad \phi^t_{\sigma,\pi}(\Sigma) := \rho^e\left(\rho^p\left(\phi^{t-1}_{\sigma,\pi}(\Sigma), \pi_{t-1}, \sigma_{t-1}\right), \pi_t\right), \quad t \in [1, T]$$

where ρ^p and ρ^e are the Riccati prediction (2.4) and update (2.5) steps, respectively.

Lemma D.1 (Vitus et al. (2012))**.** *For any $t \in [0,T]$, the t-step Riccati map is operator monotone and operator concave.*

Proof of Thm 2.2. Let (x, Σ, J) be the dominated node so that, from Definition 2.1, there exist nonnegative constants $\{\alpha_i\}_{i=1}^K$ such that:

$$\Sigma \succeq \sum_{i=1}^{K} \alpha_i \Sigma^i \quad \text{and} \quad J \geq \sum_{i=1}^{K} \alpha_i J^i.$$

Let $\sigma \in U^{T-t}$ be any admissible control sequence applied at time t to the sensor state $x = x^i,\ i = 1, \ldots, K$ with corresponding state trace $\pi \in \mathcal{X}^{T+1-t}$. From monotonicity and concavity of the Riccati map (Lemma D.1) for any $\tau \in [t, T]$:

$$\phi_{\sigma,\pi}^{\tau-t}(\Sigma) \succeq \phi_{\sigma,\pi}^{\tau-t}\left(\sum_{i=1}^{K} \alpha_i \Sigma^i\right) \succeq \sum_{i=1}^{K} \alpha_i \phi_{\sigma,\pi}^{\tau-t}\left(\Sigma^i\right).$$

Then, from monotonicity and concavity of c_t:

$$\begin{aligned} J + \sum_{\tau=t+1}^{T} c_\tau\left(\phi_{\sigma,\pi}^{\tau-t}(\Sigma), \pi_{\tau-t}\right) &\geq \left(\sum_{i=1}^{K} \alpha_i J^i\right) + \sum_{\tau=t+1}^{T} c_\tau\left(\sum_{i=1}^{K} \alpha_i \phi_{\sigma,\pi}^{\tau-t}\left(\Sigma^i\right), \pi_{\tau-t}\right) \\ &\geq \sum_{i=1}^{K} \alpha_i \left(J^i + \sum_{\tau=t+1}^{T} c_\tau\left(\phi_{\sigma,\pi}^{\tau-t}\left(\Sigma^i\right), \pi_{\tau-t}\right)\right) \\ &\geq J^{i^*} + \sum_{\tau=t+1}^{T} c_\tau\left(\phi_{\sigma,\pi}^{\tau-t}\left(\Sigma^{i^*}\right), \pi_{\tau-t}\right). \end{aligned}$$

The last inequality holds because a convex combination of scalars is lower bounded by the smallest one, i^*. The inequalities above show that, for any control sequence, the total cost obtained from the dominated node (x, Σ, J) will be bounded below by the total cost obtained one of the dominating nodes. Thus, (x, Σ, J) cannot lead to the optimal solution. □

D.3 Proof of Theorem 2.3

Notice that Thm. 2.3 is a special case of Thm. 2.4 because if $\delta = 0$, then $\zeta_t = 1$, $\forall t \in [1, T]$ and since $\eta_* < 1$, Δ_T reduces to:

$$\begin{aligned} \Delta_T &= \frac{d_y}{\lambda_*}\left(1 + \frac{\beta_*}{\lambda_*} \sum_{\tau=1}^{T-1} \eta_*^{T-\tau}\right) = \frac{d_y}{\lambda_*}\left(1 + \frac{\beta_*}{\lambda_*} \frac{\eta_*}{1-\eta_*}(1 - \eta_*^{T-1})\right) \\ &= \frac{d_y}{\lambda_*}\left(1 + \frac{\beta_*^5}{\lambda_*^5}(1 - \eta_*^{T-1})\right) \leq \frac{d_y}{\lambda_*}\left(1 + \frac{\beta_*^5}{\lambda_*^5}\right). \end{aligned}$$

□

D.4 Proof of Theorem 2.4

Lemma D.2 (Vitus et al. (2012))**.** *For any control sequence $\sigma \in U^T$, corresponding sensor state trajectory $\pi := x_0, \dots, x_T \in \mathcal{X}^{T+1}$, symmetric matrix $Q \succeq 0$, and $\epsilon \geq 0$, the directional derivative of the t-step Riccati map with $t \in [0,T]$ is:*

$$g^t_{\sigma,\pi}(\Sigma; Q) := \frac{d}{d\epsilon}\phi^t_{\sigma,\pi}(\Sigma + \epsilon Q)\bigg|_{\epsilon=0} = \left[\prod_{j=t-1}^{0} U_{j+1}\big(\phi^j_{\sigma,\pi}(\Sigma)\big)\, A_j\right] Q \left[\prod_{k=0}^{t-1} A_k^T U_{k+1}^T \left(\phi^k_{\sigma,\pi}(\Sigma)\right)\right]$$

where $U_{t+1}(\Sigma) := R_{t+1}(A_t \Sigma A_t^t + W_t)$ and $R_t(\Sigma) := I - \Sigma H_t^T \left(H_t \Sigma H_t^T + V_t\right)^{-1} H_t$.

Lemma D.3. *For any $t \in [0,T]$, $\sigma \in \mathcal{U}^T$, corresponding state trajectory $\pi \in \mathcal{X}^{T+1}$, symmetric matrices $\Sigma, Q_1, Q_2, Q \succeq 0$, and $a, b \in \mathbb{R}$*

$$g^t_{\sigma,\pi}(\Sigma; aQ_1 + bQ_2) = a g^t_{\sigma,\pi}(\Sigma; Q_1) + b g^t_{\sigma,\pi}(\Sigma; Q_2)$$

because a directional derivative is linear in the perturbation. In addition, by operator concavity of the t-step Riccati map $\phi^t_{\sigma,\pi}(\Sigma + \epsilon Q) \preceq \phi^t_{\sigma,\pi}(\Sigma) + \epsilon g^t_{\sigma,\pi}(\Sigma; Q)$.

Lemma D.4. *For all $t \in [1,T]$, $\sigma \in \mathcal{U}^{T-1}$ with corresponding state trajectory $\pi \in \mathcal{X}^T$, and symmetric $\Sigma, Q \succeq 0$, if there exist constants $\lambda > 0$ and $\beta < \infty$ such that $(W_t^{-1} + M_{t+1})^{-1} \succeq \lambda I$, $\Sigma_t := \phi^t_{\sigma,\pi}(\Sigma) \preceq \beta I$, and $A_t \Sigma_t A_t^T + W_t \preceq \beta I$, then:*

$$\operatorname{tr}\big(g^t_\pi(\Sigma; Q)\big) \leq \beta \eta^t \operatorname{tr}(\Sigma^{-1} Q), \qquad \eta := \frac{\beta^4}{(\beta^4 + \lambda^4)} < 1.$$

Proof of Lemma D.4. Note that $W_t \succeq (W_t^{-1} + M_{t+1})^{-1} \succeq \lambda I$ and

$$\begin{aligned}
\phi^1_{\sigma,\pi}(\Sigma) &\succeq \phi^1_{\sigma,\pi}(0) = (W_0^{-1} + M_1)^{-1} \succeq \lambda I \\
\phi^2_{\sigma,\pi}(\Sigma) &\succeq ((\lambda A_1 A_1^T + W_1)^{-1} + M_2)^{-1} \succeq (W_1^{-1} + M_2)^{-1} \succeq \lambda I \\
\Sigma_t := \phi^t_{\sigma,\pi}(\Sigma) &\succeq \lambda I \quad \text{for } t \in [1,T].
\end{aligned}$$

We follow the proof of (Vitus et al. 2012, Thm. 5) but our Riccati map is defined in the opposite order, i.e., prediction step first and then update step. Moore and Anderson (1980) show that the Riccati recursion can be written as:

$$\Sigma_{t+1} = U_{t+1} A_t \Sigma_t A_t^T U_{t+1}^T + U_{t+1} W_t U_{t+1}^T + S_{t+1} V_{t+1} S_{t+1}^T$$

where $U_{t+1} := R_{t+1}(A_t \Sigma_t A_t^T + W_t)$, $R_t(\Sigma) := I - K_t(\Sigma) H_t$, the Kalman gain is $K_t(\Sigma) := \Sigma H_t^T \left(H_t \Sigma H_t^T + V_t\right)^{-1}$, and $S_{t+1} := K_{t+1}(A_t \Sigma_t A_t^T + W_t)$. Also, note that $\Sigma_{t+1} = U_{t+1}(A_t \Sigma_t A_t^T + W_t)$. From the assumptions and since the matrices W_t, V_t, and $M_t := H_t^T V_t^{-1} H_t$ are symmetric positive semidefinite:

$$\begin{aligned}
\beta I \succeq \Sigma_{t+1} &\succeq U_{t+1} W_t U_{t+1}^T \\
&= \Sigma_{t+1}(A_t \Sigma_t A_t^T + W_t)^{-1} W_t (A_t \Sigma_t A_t^T + W_t)^{-1} \Sigma_{t+1} \succeq \frac{\lambda^3}{\beta^2} I
\end{aligned}$$

and $\frac{1}{\beta}I \preceq \Sigma_t^{-1}$. Let $\alpha := \lambda^3/\beta^4$ and note that $\alpha I \preceq \alpha\beta\Sigma_t^{-1}$ and thus $\Sigma_t^{-1}-\alpha I \succeq (1-\alpha\beta)\Sigma_t^{-1}$. Since from the inequalities above, $\alpha\beta = \lambda^3/\beta^3 \leq 1$, it follows that for $t = 0, \dots, T-1$:

$$\begin{aligned} U_{t+1}A_t(\Sigma_t^{-1} - \alpha I)^{-1}A_t^T U_{t+1}^T &\preceq (1-\alpha\beta)^{-1}U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T \\ &= U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T + \left(\frac{1}{1-\alpha\beta} - 1\right) U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T \\ &\preceq U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T + \frac{\alpha\beta}{1-\alpha\beta}\left(\beta - \frac{\lambda^3}{\beta^2}\right) I \\ &= U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T + \frac{\lambda^3}{\beta^2} I \end{aligned}$$

where the second to last step follows from:

$$\begin{aligned} \beta I \succeq \Sigma_{t+1} &= U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T + U_{t+1}W_t U_{t+1}^T + S_{t+1}V_{t+1}S_{t+1}^T \\ &\succeq U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T + U_{t+1}W_t U_{t+1}^T \succeq U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T + \frac{\lambda^3}{\beta^2} I. \end{aligned}$$

Note also that $\Sigma_{t+1} - U_{t+1}A_t\Sigma_t A_t^T U_{t+1}^T \succeq U_{t+1}W_t U_{t+1}^T \succeq \frac{\lambda^3}{\beta^2}I$ and therefore:

$$\Sigma_{t+1} - U_{t+1}A_t(\Sigma_t^{-1} - \alpha I)^{-1}A_t^T U_{t+1}^T \succeq \Sigma_{t+1} - U_{t+1}A_t\Sigma_t^{-1}A_t^T U_{t+1}^T - \frac{\lambda^3}{\beta^2} I \succeq 0.$$

Applying a Schur complement to the last result shows that:

$$\Sigma_t^{-1} - U_{t+1}^T A_t^T \Sigma_{t+1}^{-1} A_t U_{t+1} \succeq \frac{\lambda^3}{\beta^4} I.$$

From here we follow the steps of (Vitus et al. 2012, Thm. 5), where in our case $\alpha := \lambda^3/\beta^4$, $\hat{A}_t := U_{t+1}A_t$, and $\underline{\lambda}_W := \lambda$. Since $Q \succeq 0$ is symmetric, we can decompose it as $Q = \sum_{l=1}^n \lambda_Q^l q_l q_l^T$ and let $\xi^{(l)}(0) = \sqrt{\lambda_Q^l} q_l$. □

Lemma D.5. *For $t \in [1, T]$, $\epsilon \geq 0$, $\delta \geq 0$, the reduced tree $\mathcal{T}_t^{\epsilon,\delta}$ contains a set of nodes $\{(x_t^i, \Sigma_t^i) \mid i = 1, \dots, K\}$ such that:*

$$d_{\mathcal{X}}(x_t^*, x_t^i) \leq \sum_{\tau=0}^{t-1} L_f^\tau \delta, \quad \forall i, \tag{D.2}$$

$$\Sigma_t^* + \epsilon\left(\Gamma_t I + \sum_{\tau=1}^{t-1} \Gamma_\tau g_*^{\tau:t}(\Sigma_\tau^*; I)\right) \succeq \Gamma_t \sum_{i=1}^{K} \alpha_i \Sigma_t^i + \sum_{\tau=1}^{t-1} \Gamma_\tau (1-\gamma_\tau)\phi_*^{\tau:t}(0), \tag{D.3}$$

where $\phi_*^{\tau:t}$ is the $(t-\tau)$-step Riccati map, associated with the optimal control sequence $\sigma_{\tau:(t-1)}^*$ and the optimal state trajectory $x_\tau^*, \dots, x_t^*$, $g_*^{\tau:t}$ is its directional derivative, defined in Lemma D.2, and

$$0 < \gamma_t := (1 + \textstyle\sum_{s=1}^t L_f^s L_m \delta)^{-1} \leq 1, \quad \textit{and} \quad \Gamma_t := \textstyle\prod_{s=1}^{t-1} \gamma_s.$$

Proof of Lemma D.5. We proceed by induction.
Base Case: At time 1, (D.2), (D.3) follow from ϵ-algebraic redundancy (Def. 2.2) and trajectory δ-crossing (Def. 2.3).
Hypothesis: Suppose that (D.2) and (D.3) hold for some set $\{(x_t^j, \Sigma_t^j) \mid j = 1, \dots, K\}$ of nodes in $\mathcal{T}_t^{\epsilon,\delta}$.
Induction: At time $t+1$, there are sets $\{(x_{t+1}^{ji}, \Sigma_{t+1}^{ji})\}_{i=1}^{K_j}$ in $\mathcal{T}_{t+1}^{\epsilon,\delta}$ corresponding to each node j from time t and satisfying

$$d_{\mathcal{X}}(x_{t+1}^j, x_{t+1}^{ji}) \leq \delta \quad \text{and} \quad \Sigma_{t+1}^j + \epsilon I \succeq \textstyle\sum_{i=1}^{K_j} \alpha_{ji} \Sigma_{t+1}^{ji}. \tag{D.4}$$

From Lemma D.2 for every $\tau = 1, \dots, t$:

$$g_*^{t:(t+1)}\left(\Sigma_t^*; g_*^{\tau:t}(\Sigma_\tau^*; I)\right) = g_*^{\tau:(t+1)}(\Sigma_\tau^*; I).$$

From this and Lemma D.3:

$$\begin{aligned}\Sigma_{t+1}^* + \epsilon \sum_{\tau=1}^{t} \Gamma_\tau g_*^{\tau:(t+1)}(\Sigma_\tau^*; I) &= \phi_*^{t:(t+1)}(\Sigma_t^*) + \epsilon g_*^{t:(t+1)}\left(\Sigma_t^*; \sum_{\tau=1}^{t} \Gamma_\tau g_*^{\tau:t}(\Sigma_\tau^*; I)\right) \\ &\succeq \phi_*^{t:(t+1)}\left(\Sigma_t^* + \epsilon \sum_{\tau=1}^{t-1} \Gamma_\tau g_*^{\tau:t}(\Sigma_\tau^*; I) + \epsilon \Gamma_t I\right)\end{aligned}$$

Note that $\sum_{\tau=1}^{t-1} \Gamma_\tau(1-\gamma_\tau) + \Gamma_t = 1$. Thus, the terms $(1-\gamma_1), \gamma_1(1-\gamma_2), \dots, \Gamma_{t-1}(1-\gamma_{t-1}), \Gamma_t$ are the coefficients of a convex combination. Using the hypothesis and monotonicity and concavity of the Riccati map (Lemma D.1):

$$\begin{aligned}&\phi_*^{t:(t+1)}\left(\Sigma_t^* + \epsilon \sum_{\tau=1}^{t-1} \Gamma_\tau g_*^{\tau:t}(\Sigma_\tau^*; I) + \epsilon \Gamma_t I\right) \\ &\qquad \succeq \phi_*^{t:(t+1)}\left(\Gamma_t \sum_{j=1}^{K} \alpha_j \Sigma_t^j + \sum_{\tau=1}^{t-1} \Gamma_\tau (1-\gamma_\tau) \phi_*^{\tau:t}(0)\right) \\ &\qquad \succeq \Gamma_t \sum_{j=1}^{K} \alpha_j \phi_*^{t:(t+1)}(\Sigma_t^j) + \sum_{\tau=1}^{t-1} \Gamma_\tau (1-\gamma_\tau) \phi_*^{\tau:(t+1)}(0).\end{aligned}$$

By hypothesis, $d_{\mathcal{X}}(x_t^*, x_t^j) \leq \sum_{\tau=0}^{t-1} L_f^\tau \delta$, and from the continuity assumption on the Riccati map (Assumption 2.2):

$$\phi_*^{t:(t+1)}(\Sigma_t^j) \succeq \gamma_t \Sigma_{t+1}^j + (1-\gamma_t) \phi_*^{t:(t+1)}(0).$$

The nodes $\{(x_{t+1}^j, \Sigma_{t+1}^j)\}$ might not be in $\mathcal{T}_{t+1}^{\epsilon,\delta}$ but from (D.4):

$$\begin{aligned}\phi_*^{t:(t+1)}(\Sigma_t^j) + \gamma_t \epsilon I &\succeq \gamma_t \left(\Sigma_{t+1}^j + \epsilon I\right) + (1-\gamma_t) \phi_*^{t:(t+1)}(0) \\ &\succeq \gamma_t \sum_{i=1}^{K_j} \alpha_{ji} \Sigma_{t+1}^{ji} + (1-\gamma_t) \phi_*^{t:(t+1)}(0).\end{aligned}$$

Combining the previous results, we have:

$$\begin{aligned}
&\Sigma^*_{t+1} + \epsilon \sum_{\tau=1}^{t} \Gamma_\tau g_*^{\tau:(t+1)}(\Sigma^*_\tau; I) + \epsilon \Gamma_{t+1} I \\
&\succeq \Gamma_t \sum_{j=1}^{K} \alpha_j \left(\phi_*^{t:(t+1)}(\Sigma^j_t) + \gamma_t \epsilon I \right) + \sum_{\tau=1}^{t-1} \Gamma_\tau (1-\gamma_\tau) \phi_*^{\tau:(t+1)}(0) \\
&\succeq \Gamma_t \sum_{j=1}^{K} \alpha_j \left(\gamma_t \sum_{i=1}^{K_j} \alpha_{ji} \Sigma^{ji}_{t+1} + (1-\gamma_t) \phi_*^{t:(t+1)}(0) \right) + \sum_{\tau=1}^{t-1} \Gamma_\tau (1-\gamma_\tau) \phi_*^{\tau:(t+1)}(0) \\
&= \Gamma_{t+1} \sum_{j=1}^{K} \sum_{i=1}^{K_j} \alpha_j \alpha_{ji} \Sigma^{ji}_{t+1} + \sum_{\tau=1}^{t} \Gamma_\tau (1-\gamma_\tau) \phi_*^{\tau:(t+1)}(0).
\end{aligned}$$

Thus, the set $\bigcup_{j=1}^{K} \bigcup_{i=1}^{K_j} \{(x^{ji}_{t+1}, \Sigma^{ji}_{t+1})\}$ satisfies (D.3) at time $t+1$. It also satisfies (D.2) because of (D.4) and the motion model continuity (Assumption 2.1). □

Proof of Theorem 2.4. We use the notation from Lemma D.5 so that $\Gamma_T = \zeta_T^{-1}$. Further, let $J(\cdot) := \log\det(\cdot)$ and $G := \Gamma_T I + \sum_{\tau=1}^{T-1} \Gamma_\tau g_*^{\tau:T}(\Sigma^*_\tau; I)$. By monotonicity of $J(\cdot)$ and the result (D.3) in Lemma D.5:

$$J(\Sigma^*_T + \epsilon G) \geq J\left(\Gamma_T \sum_{i=1}^{K} \alpha_i \Sigma^i_T + \sum_{\tau=1}^{T-1} \Gamma_\tau (1-\gamma_\tau) \phi_*^{\tau:T}(0) \right)$$

for some set of nodes $\{(x^i_T, \Sigma^i_T) \mid i = 1, \dots, K\}$ in the reduced tree $\mathcal{T}^{\epsilon,\delta}_T$. Note that $\sum_{\tau=1}^{t-1} \Gamma_\tau (1-\gamma_\tau) + \Gamma_t = 1$ and, as shown in the proof of Lemma D.4, $\phi_*^{\tau:T}(0) \succeq \lambda_* I$. Then, by concavity and monotonicity of $J(\cdot)$:

$$\begin{aligned}
J(\Sigma^*_T + \epsilon G) &\geq \Gamma_T \sum_{i=1}^{K} \alpha_i J(\Sigma^i_T) + \sum_{\tau=1}^{T-1} \Gamma_\tau (1-\gamma_\tau) J\left(\phi_*^{\tau:T}(0)\right) \qquad \text{(D.5)} \\
&\geq \Gamma_T J(\Sigma^{i^*}_T) + \sum_{\tau=1}^{T-1} \Gamma_\tau (1-\gamma_\tau) J(\lambda_* I) \geq \Gamma_T J^{\epsilon,\delta}_T + (1-\Gamma_T) J(\lambda_* I).
\end{aligned}$$

The second inequality holds because a convex combination of scalars is lower bounded by the smallest one i^*. The last inequality holds because $\pi^{\epsilon,\delta}$ is the optimal path in the reduced tree. Next, by concavity of $\log\det(\cdot)$:

$$\begin{aligned}
J(\Sigma^*_T + \epsilon G) &\leq J(\Sigma^*_T) + \epsilon \frac{d}{d\epsilon} J\left(\Sigma^*_T + \epsilon G \right)\Bigg|_{\epsilon=0} \\
&= J^*_T + \epsilon \operatorname{tr}\left((\Sigma^*_T)^{-1} G \right) \leq J^*_T + \epsilon \frac{1}{\lambda_*} \operatorname{tr}(G). \qquad \text{(D.6)}
\end{aligned}$$

From Lemma D.4 and since $\operatorname{tr}((\Sigma_\tau^*)^{-1}) \leq d_y/\lambda_*$:

$$\begin{aligned}
\operatorname{tr}(G) &= \Gamma_T \operatorname{tr}(I_{d_y}) + \sum_{\tau=1}^{T-1} \Gamma_\tau \operatorname{tr}\Big(g_*^{\tau:T}(\Sigma_\tau^*; I_{d_y})\Big) \\
&\leq d_y \Gamma_T + \sum_{\tau=1}^{T-1} \Gamma_\tau \beta_* \eta_*^{T-\tau} \operatorname{tr}((\Sigma_\tau^*)^{-1}) \leq \lambda_* \Gamma_T \Delta_T
\end{aligned} \tag{D.7}$$

Finally, by combining (D.5), (D.6), and (D.7) we get:

$$\begin{aligned}
\Gamma_T J_T^{\epsilon,\delta} + (1-\Gamma_T) J(\lambda_* I) &\leq J_T^* + \epsilon \Gamma_T \Delta_T \\
0 \leq \Gamma_T (J_T^{\epsilon,\delta} - J_T^*) &\leq (1-\Gamma_T)(J_T^* - J(\lambda_* I)) + \epsilon \Gamma_T \Delta_T.
\end{aligned}$$

Multiplying by $\zeta_T = \Gamma_T^{-1}$ gives the result in (2.8). □

D.5 Proof of Proposition 2.5

Consider the left-hand side:

$$L := \begin{bmatrix} (W_t^d)^{-1/2} & 0 & 0 \\ D_t(A_t^d)^{-1} & D_t(A_t^d)^{-1} & F_t \\ 0 & 0 & G_t \\ 0 & V_{t+1}^{-1/2} H_{t+1}^d & V_{t+1}^{-1/2} H_{t+1}^s \end{bmatrix}$$

where H_{t+1} has also been decomposed into its static and dynamic parts. Let $E_t := (A_t^d)^{-T} D_t^T D_t (A_t^d)^{-1}$ and $B_t := (W_t^d)^{-1} + E_t$. Then, $L^T L$ equals:

$$\begin{bmatrix} B_t & E_t & (A_t^d)^{-T} D_t^T F_t \\ E_t & E_t + (H_{t+1}^d)^T V_{t+1}^{-1} H_{t+1}^d & (A_t^d)^{-T} D_t^T F_t + (H_{t+1}^d)^T V_{t+1}^{-1} H_{t+1}^s \\ F_t^T D_t (A_t^d)^{-1} & F_t^T D_t (A_t^d)^{-1} + (H_{t+1}^s)^T V_{t+1}^{-1} H_{t+1}^d & F_t^T F_t + G_t^T G_t + (H_{t+1}^s)^T V_{t+1}^{-1} H_{t+1}^s \end{bmatrix}$$

Similarly, the right-hand side is:

$$\begin{aligned}
R &:= Q \begin{bmatrix} B_t^{1/2} & B_t^{-1/2} E_t & B_t^{-1/2} (A_t^d)^{-T} D_t^T F_t \\ 0 & D_{t+1} & F_{t+1} \\ 0 & 0 & G_{t+1} \\ 0 & 0 & 0 \end{bmatrix} \\
R^T R &= \begin{bmatrix} B_t & E_t & (A_t^d)^{-T} D_t^T F_t \\ E_t & E_t^T B_t^{-1} E_t + D_{t+1}^T D_{t+1} & E_t^T B_t^{-1} (A_t^d)^{-1} D_t^T F_t + D_{t+1}^T F_{t+1} \\ F_t^T D_t (A_t^d)^{-1} & * & P_t + F_{t+1}^T F_{t+1} + G_{t+1}^T G_{t+1} \end{bmatrix}
\end{aligned}$$

where $P_t := F_t^T D_t (A_t^d)^{-1} B_t^{-1} (A_t^d)^{-T} D_t^T F_t$ and $*$ is such that the matrix is symmetric. Thus, to show that $L^T L = R^T R$ it suffices to verify that:

$$\begin{aligned}
E_t + (H_{t+1}^d)^T V_{t+1}^{-1} H_{t+1}^d &= E_t^T B_t^{-1} E_t + D_{t+1}^T D_{t+1} \\
(A_t^d)^{-T} D_t^T F_t + (H_{t+1}^d)^T V_{t+1}^{-1} H_{t+1}^s &= E_t^T B_t^{-1} (A_t^d)^{-1} D_t^T F_t + D_{t+1}^T F_{t+1} \\
F_t^T F_t + G_t^T G_t + (H_{t+1}^s)^T V_{t+1}^{-1} H_{t+1}^s &= P_t + F_{t+1}^T F_{t+1} + G_{t+1}^T G_{t+1}
\end{aligned} \tag{D.8}$$

The any t the covariance matrix satisfies:

$$\Sigma_t = \begin{bmatrix} \Sigma_t^d & \Sigma_t^{ds} \\ (\Sigma_t^{ds})^T & \Sigma_t^s \end{bmatrix} = \Omega_t^{-1} = (C_t^T C_t)^{-1}, \qquad C_t = \begin{bmatrix} D_t & F_t \\ 0 & G_t \end{bmatrix}$$

$$\begin{aligned} \Sigma_t^d &= \left(D_t^T D_t - D_t^T F_t (F_t^T F_t + G_t^T G_t)^{-1} F_t^T D_t\right)^{-1} \\ \Sigma_t^{ds} &= -D_t^{-1} F_t \left(G_t^T G_t\right)^{-1} \qquad \Sigma^s = \left(G_t^T G_t\right)^{-1} \\ D_t^T D_t &= \left(\Sigma_t^d - (\Sigma_t^{ds})^T (\Sigma_t^s)^{-1} \Sigma_t^{ds}\right)^{-1} \end{aligned} \tag{D.9}$$

From the Kalman filter equations (Appendix C.1):

$$\begin{aligned} \Omega_{t+1} &= \begin{bmatrix} D_{t+1}^T D_{t+1} & D_{t+1}^T F_{t+1} \\ F_{t+1}^T D_{t+1} & F_{t+1}^T F_{t+1} + G_{t+1}^T G_{t+1} \end{bmatrix} \\ &= \begin{bmatrix} A_t^d \Sigma_t^d (A_t^d)^T + W_t^d & A_t^d \Sigma_t^{ds} \\ (\Sigma_t^{ds})^T (A_t^d)^T & \Sigma_t^s \end{bmatrix}^{-1} + \begin{bmatrix} (H_{t+1}^d)^T V_{t+1}^{-1} H_{t+1}^d & (H_{t+1}^d)^T V_{t+1} H_{t+1}^s \\ (H_{t+1}^s)^T V_{t+1} H_{t+1}^d & (H_{t+1}^s)^T V_{t+1} H_{t+1}^s \end{bmatrix} \end{aligned}$$

Inverting the first matrix above and using the relations in (D.9), we get:

$$\begin{aligned} D_{t+1}^T D_{t+1} &= (A_t^d D_t^T D_t (A_t^d)^T + W_t^d)^{-1} + (H_{t+1}^d)^T V_{t+1}^{-1} H_{t+1}^d \\ D_{t+1}^T F_{t+1} &= (A_t^d D_t^T D_t (A_t^d)^T + W_t^d)^{-1} A_t^d D_t^{-1} F_t + (H_{t+1}^d)^T V_{t+1} H_{t+1}^s \\ F_{t+1}^T F_{t+1} + G_{t+1}^T G_{t+1} &= \left(\Sigma_t^s - \Sigma_t^{dsT} A_t^{dT} (A_t^d \Sigma_t^d A_t^{dT} + W_t^d)^{-1} A_t^d \Sigma_t^{ds}\right)^{-1} + H_{t+1}^{sT} V_{t+1} H_{t+1}^s \end{aligned}$$

Applying the matrix inversion lemma to the equations above verifies (D.8). □

D.6 Proof of Theorem 2.6

Lemma D.6. *Let $\mathcal{F} \subseteq \{1, \dots, n\}$, $y_{\mathcal{F}} := \{y_{i,1:T} \mid i \in \mathcal{F}\}$, and $z_{\mathcal{F}} := \{z_{i,1:T} \mid i \in \mathcal{F}\}$. The mutual information $g(\mathcal{F}) := \mathbb{I}(y_{\mathcal{F}}; z_{\mathcal{F}})$ between the target states $y_{\mathcal{F}}$ and the measurement set $z_{\mathcal{F}}$ viewed as a function of the set of sensors $\mathcal{F}$ is submodular. In other words, given sets $\mathcal{A} \subseteq \mathcal{B} \subseteq \{1, \dots, n\}$ and $\mathcal{C} \subseteq \{1, \dots, n\} \setminus \mathcal{B}$, the following is satisfied: $g(\mathcal{A} \cup \mathcal{C}) - g(\mathcal{A}) \geq g(\mathcal{B} \cup \mathcal{C}) - g(\mathcal{B})$.*

Proof of Lemma D.6. The proof follows the steps of (Williams 2007, Lemma 2.1) but to handle the varying dimension of $y_{\mathcal{F}}$ we use that for $\mathcal{A} \subseteq \mathcal{B}$

$$\mathbb{I}(y_{\mathcal{B}}; z_{\mathcal{A}}) = \mathbf{h}(z_{\mathcal{A}}) - \mathbf{h}(z_{\mathcal{A}} \mid y_{\mathcal{B}}) = \mathbf{h}(z_{\mathcal{A}}) - \mathbf{h}(z_{\mathcal{A}} \mid y_{\mathcal{A}}) = \mathbb{I}(y_{\mathcal{A}}; z_{\mathcal{A}}),$$

since when conditioned on $y_{\mathcal{A}}$, the measurements $z_{\mathcal{A}}$ from known sensor states $x_{\mathcal{A}} := \{x_{i,1:T} \mid i \in \mathcal{A}\}$ are independent of all other target states. □

Proof of Theorem 2.6. Let the measurement sets obtained by μ^* and μ^c be $z_{1:T}^*$ and $z_{1:T}^c$, respectively. Then,

$$\begin{aligned}
\mathbb{I}(y_{1:T}; z^*_{1:T}) &\overset{(a)}{\leq} \mathbb{I}(y_{1:T}; z^*_{1:T}) + \sum_{i=1}^{n}\Bigg(\mathbb{I}(y_{1:T}; z^c_{1:i,1:T}, z^*_{i+1:n,1:T}) \\
&\qquad\qquad - \mathbb{I}(y_{1:i-1,1:T}, y_{i+1:n,1:T}; z^c_{1:i-1,1:T}, z^*_{i+1:n,1:T})\Bigg) \\
&= \mathbb{I}(y_{1:T}; z^c_{1:T}) + \sum_{i=1}^{n}\Bigg(\mathbb{I}(y_{1:T}; z^c_{1:i-1,1:T}, z^*_{i:n,1:T}) \\
&\qquad\qquad - \mathbb{I}(y_{1:i-1,1:T}, y_{i+1:n,1:T}; z^c_{1:i-1,1:T}, z^*_{i+1:n,1:T})\Bigg) \\
&\overset{\text{Lemma D.6}}{\leq} \mathbb{I}(y_{1:T}; z^c_{1:T}) + \sum_{i=1}^{n}\Bigg(\mathbb{I}(y_{1:i,1:T}; z^c_{1:i-1,1:T}, z^*_{i,1:T}) - \mathbb{I}(y_{1:i-1,1:T}; z^c_{1:i-1,1:T})\Bigg) \\
&= \mathbb{I}(y_{1:T}; z^c_{1:T}) + \mathbb{I}(y_{1:T}; z^c_{1:n-1,1:T}, z^*_{n,1:T}) \qquad \text{(D.10)} \\
&\qquad + \sum_{i=1}^{n-1}\left(\mathbb{I}(y_{1:i,1:T}; z^c_{1:i-1,1:T}, z^*_{i,1:T}) - \mathbb{I}(y_{1:i,1:T}; z^c_{1:i,1:T})\right)
\end{aligned}$$

where (a) follows from the non-decreasing property of mutual information. By definition of the coordinate descent policy

$$-\mathbb{I}(y_{1:i,1:T}; z^c_{1:i-1,1:T}, z^*_{i,1:T}) \geq -\mathbb{I}(y_{1:i,1:T}; z^c_{1:i,1:T}),\ \forall i$$

and from (D.10), $J^*_T = -\mathbb{I}(y_{1:T}; z^*_{1:T}) \geq J^c_T + J^c_T + 0$. □

D.7 Proof of Corollary 2.7

Lemma D.7. *Given the motion and observation models in* (2.1) *and a deterministic state sequence* $x_{0:T}$*, the following relationship between mutual information and conditional entropy holds:*

$$\mathbf{h}(y_{1:T}) - \sum_{t=1}^{T}\mathbf{h}(y_t \mid z_{1:t}) \leq \mathbb{I}(y_{1:T}; z_{1:T}) \leq \sum_{t=1}^{T} h(y_t) - \sum_{t=1}^{T}\mathbf{h}(y_t \mid z_{1:t})$$

Proof of Lemma D.7. $\mathbf{h}(y_{1:T}) - \sum_{t=1}^{T}\mathbf{h}(y_t \mid z_{1:t}) \overset{(a)}{\leq} \mathbf{h}(y_{1:T}) - \sum_{t=1}^{T}\mathbf{h}(y_t \mid z_{1:t}, y_{t-1})$
$\overset{(b)}{=\!=} \mathbf{h}(y_{1:T}) - \mathbf{h}(y_{1:T} \mid z_{1:T}) = \mathbb{I}(y_{1:T}; z_{1:T}) = \sum_{t=1}^{t}[\mathbf{h}(y_t \mid z_{1:t-1}) - h(y_t \mid z_{1:t})]$
$\overset{(c)}{\leq} \sum_{t=1}^{T}\mathbf{h}(y_t) - \sum_{t=1}^{T}\mathbf{h}(y_t \mid z_{1:t})$
where $(a), (c)$ hold because conditioning decreases entropy, and (b) is due to the entropy chain rule. □

Proof of Corollary 2.7. From Lemma D.7 and (D.10), it follows that

$$
\begin{aligned}
\mathbf{h}(y_{1:T}) - J_T^* \leq& \sum_{t=1}^{T} \mathbf{h}(y_t) - J_T^c + \sum_{t=1}^{T} \mathbf{h}(y_t) - \sum_{t=1}^{T} \mathbf{h}(y_t \mid z_{1:n-1,1:t}^c, z_{n,1:t}^*) \\
&+ \sum_{i=1}^{n-1} \left(\left(\sum_{t=1}^{T} \mathbf{h}(y_{1:i,t}) \right) - \mathbf{h}(y_{1:i,1:T}) \right) \\
&+ \sum_{i=1}^{n-1} \left(\sum_{t=1}^{T} \mathbf{h}(y_{1:i,t} \mid z_{1:i,1:t}^c) - \sum_{t=1}^{T} \mathbf{h}(y_{1:i,t} \mid z_{1:i-1,1:t}^c, z_{i,1:t}^*) \right) \\
&\leq 2 \sum_{t=1}^{T} \mathbf{h}(y_t) - 2J_T^c + \left(\sum_{i=1}^{n-1} \left(\sum_{t=1}^{T} \mathbf{h}(y_{1:i,t}) \right) - \mathbf{h}(y_{1:i,1:T}) \right)
\end{aligned}
$$

□

D.8 Proof of Theorem 2.10

Define the following:

$$
\begin{aligned}
\omega_t &:= \begin{bmatrix} \omega_{1,t}^T & \dots & \omega_{n,t}^T \end{bmatrix}^T & \Omega_t &:= \begin{bmatrix} \Omega_{1,t}^T & \dots & \Omega_{n,t}^T \end{bmatrix}^T \\
M_i &:= H_i(x_i)^T V_i^{-1}(x_i) H_i(x_i) & M &:= \begin{bmatrix} M_1^T & \dots & M_n^T \end{bmatrix}^T \\
\zeta_t &:= \begin{bmatrix} H_1 V_1^{-T} v_{1,t}^T & \dots & H_n V_n^{-T} v_{n,t}^T \end{bmatrix}^T.
\end{aligned}
$$

The update equations of the filter (2.25) in matrix form are:

$$
\begin{aligned}
\omega_{t+1} &= (\mathcal{K} \otimes I_{d_y}) \omega_t + My + \zeta_t, \\
\Omega_{t+1} &= (\mathcal{K} \otimes I_{d_y}) \Omega_t + M,
\end{aligned} \tag{D.11}
$$

where $\mathcal{K} = [\kappa_{ij}]$ with $\kappa_{ij} = 0$ if $j \notin \mathsf{N}_i \cup \{i\}$ is a stochastic matrix. The solutions of the two linear dynamical systems are:

$$
\begin{aligned}
\omega_t &= (\mathcal{K} \otimes I_{d_y})^t \omega_0 + \sum_{\tau=0}^{t-1} (\mathcal{K} \otimes I_{d_y})^{t-1-\tau} \Big(My + \zeta_\tau \Big), \\
\Omega_t &= (\mathcal{K} \otimes I_{d_y})^t \Omega_0 + \sum_{\tau=0}^{t-1} (\mathcal{K} \otimes I_{d_y})^{t-1-\tau} M.
\end{aligned}
$$

Looking at the i-th components again, we have:

$$
\begin{aligned}
\frac{\omega_{it}}{t+1} &= \frac{1}{t+1} \sum_{j=1}^{n} [\mathcal{K}^t]_{ij} \omega_{j0} + \frac{1}{t+1} \sum_{\tau=0}^{t-1} \sum_{j=1}^{n} [\mathcal{K}^{t-\tau-1}]_{ij} (M_j y + H_j^T V_j^{-1} v_{j,\tau}), \\
\frac{\Omega_{it}}{t+1} &= \frac{1}{t+1} \sum_{j=1}^{n} [\mathcal{K}^t]_{ij} \Omega_{j0} + \frac{1}{t+1} \sum_{\tau=0}^{t-1} \sum_{j=1}^{n} [\mathcal{K}^{t-\tau-1}]_{ij} M_j.
\end{aligned}
$$

Define the following to simplify the notation:

$$\begin{aligned}
g_{it} &:= \tfrac{1}{t+1}\textstyle\sum_{j=1}^{n}\big[\mathcal{K}^{t}\big]_{ij}\omega_{j0}, & G_{it} &:= \tfrac{1}{t+1}\textstyle\sum_{j=1}^{n}\big[\mathcal{K}^{t}\big]_{ij}\Omega_{j0},\\
b_{it} &:= g_{it} - G_{it}y, & B_{it} &:= \tfrac{1}{t+1}\Omega_{it}, \qquad\qquad \text{(D.12)}\\
\phi_{it} &:= \tfrac{1}{t+1}\textstyle\sum_{\tau=0}^{t-1}\sum_{j=1}^{n}\big[\mathcal{K}^{t-\tau-1}\big]_{ij}H_j^T V_j^{-1} v_{j,\tau}, & C_{it} &:= \tfrac{1}{t+1}\textstyle\sum_{\tau=0}^{t-1}\sum_{j=1}^{n}\big[\mathcal{K}^{t-\tau-1}\big]_{ij}M_j.
\end{aligned}$$

With the shorthand notation:

$$\frac{\omega_{it}}{t+1} = g_{it} + \phi_{it} + C_{it}y, \qquad B_{it} = \frac{\Omega_{it}}{t+1} = G_{it} + C_{it}, \tag{D.13}$$

where ϕ_{it} is the only random quantity. Its mean is zero because the measurement noise is zero mean, while its covariance is:

$$\begin{aligned}
\mathbb{E}[\phi_{it}\phi_{it}^T] &= \frac{1}{(t+1)^2}\mathbb{E}\Bigg[\Bigg(\sum_{\tau=0}^{t-1}\sum_{j=1}^{n}\big[\mathcal{K}^{t-\tau-1}\big]_{ij}H_j^T V_j^{-1} v_{j,\tau}\Bigg)\Bigg(\sum_{s=0}^{t-1}\sum_{\eta=1}^{n}\big[\mathcal{K}^{t-s-1}\big]_{i\eta}H_\eta^T V_\eta^{-1} v_{\eta,s}\Bigg)^T\Bigg]\\
&= \frac{1}{(t+1)^2}\sum_{j=1}^{n}\sum_{\tau=0}^{t-1}\big[\mathcal{K}^{t-\tau-1}\big]_{ij}^2 H_j^T V_j^{-1}\mathbb{E}[v_{j,\tau}v_{j,\tau}^T]V_j^{-1}H_j\\
&= \frac{1}{(t+1)^2}\sum_{j=1}^{n}\sum_{\tau=0}^{t-1}\big[\mathcal{K}^{t-\tau-1}\big]_{ij}^2 M_j \preceq \frac{1}{t+1}C_{it},
\end{aligned} \tag{D.14}$$

where the second equality uses the fact that $v_{j,\tau}$ and $v_{\eta,s}$ are independent unless the indices coincide, i.e., $\mathbb{E}[v_{j,\tau}v_{\eta,s}^T] = \delta_{\tau s}\delta_{j\eta}V_j$. The Löwner ordering inequality in the last step uses that $0 \le \big[\mathcal{K}^{t-\tau-1}\big]_{ij} \le 1$ and $M_j \succeq 0$.

Since G is connected, $\mathcal{K}$ corresponds to the transition matrix of an aperiodic irreducible Markov chain with a unique stationary distribution π so that $\mathcal{K}^t \to \pi\mathbf{1}^T$ with $\pi_j > 0$. This implies that, as $t \to \infty$, the numerators of g_{it} and G_{it} remain bounded and therefore $g_{it} \to 0$ and $G_{it} \to 0$. Since Cesáro means preserve convergent sequences and their limits:

$$\frac{1}{t+1}\sum_{\tau=0}^{t-1}\big[\mathcal{K}^{t-\tau-1}\big]_{ij} \to \pi_j, \quad \forall i,$$

which implies that $C_{it} \to \sum_{j=1}^{n}\pi_j M_j$. The full-rank assumption on $\big[H_1^T \quad \dots \quad H_n^T\big]^T$ and $\pi_j > 0$ guarantee that $\sum_{j=1}^{n}\pi_j M_j$ is positive definite. Finally, consider the mean squared error:

$$\begin{aligned}
\mathbb{E}\big[(\hat{y}_{i,t} - y)^T(\hat{y}_{i,t} - y)\big] &= \mathbb{E}\left[\left\|\left(\frac{\Omega_{it}}{t+1}\right)^{-1}\frac{\omega_{it}}{t+1} - \left(\frac{\Omega_{it}}{t+1}\right)^{-1}\left(\frac{\Omega_{it}}{t+1}\right)y\right\|_2^2\right]\\
&= \mathbb{E}\left[\big\|B_{it}^{-1}\big(g_{it} + C_{it}y + \phi_{it} - (G_{it} + C_{it})y\big)\big\|_2^2\right]\\
&= \mathbb{E}\left[\|B_{it}^{-1}(b_{it} + \phi_{it})\|_2^2\right]\\
&= \mathbb{E}\left[b_{it}^T B_{it}^{-T}B_{it}^{-1}b_{it} + 2b_{it}^T B_{it}^{-T}B_{it}^{-1}\phi_{it} + \phi_{it}^T B_{it}^{-T}B_{it}^{-1}\phi_{it}\right]\\
&\overset{(a)}{=} b_{it}^T B_{it}^{-T}B_{it}^{-1}b_{it} + \operatorname{tr}\left(B_{it}^{-1}\mathbb{E}[\phi_{it}\phi_{it}^T]B_{it}^{-T}\right)\\
&\overset{(b)}{\le} b_{it}^T B_{it}^{-T}B_{it}^{-1}b_{it} + \frac{1}{t+1}\operatorname{tr}\left(B_{it}^{-1}C_{it}B_{it}^{-T}\right) \to 0,
\end{aligned}$$

where (a) holds because the first term is deterministic, while the cross term contains $\mathbb{E}[\phi_{it}] = 0$. Inequality (b) follows from (D.14). In the final step, as shown before $B_{it}^{-1} \to \left(\sum_{j=1}^{n} \pi_j M_j\right)^{-1}$ and $C_{it} \to \sum_{j=1}^{n} \pi_j M_j \succ 0$ remain bounded, while $b_{it} \to 0$ and $\frac{1}{t+1} \to 0$. □

D.9 Proof of Theorem 2.11

Define the *generalized (matrix-weighted) degree matrix* $D \in \mathbb{R}^{nd_x \times nd_x}$ of graph G as the block-diagonal matrix with $D_{ii} := \sum_{j \in \mathsf{N}_i} \mathcal{E}_{ij}^{-1}$. Since $\mathcal{E}_{ij} \succ 0$ for all $\{i,j\} \in E$, the generalized degree matrix is positive definite, $D \succ 0$. Define also the *generalized adjacency matrix* $\mathsf{A} \in \mathbb{R}^{nd_x \times nd_x}$ as follows:

$$\mathsf{A}_{ij} := \begin{cases} \mathcal{E}_{ij}^{-1} & \text{if } \{i,j\} \in E, \\ 0 & \text{else.} \end{cases}$$

The *generalized Laplacian* and the *generalized signless Laplacian* of G are defined as $L := D - \mathsf{A}$ and $|L| := D + \mathsf{A}$, respectively. Further, let $R := (B \otimes I_{d_x})^T \in \mathbb{R}^{md_x \times nd_x}$ and define the block-diagonal matrix $\mathcal{E} \in \mathbb{R}^{md_x \times md_x}$ with blocks $\mathcal{E}_{ij}$ for $\{i,j\} \in E$. It is straightforward to verify that $L = R^T \mathcal{E}^{-1} R \succeq 0$ and $|L| = (|B| \otimes I_{d_x}) \mathcal{E}^{-1} (|B| \otimes I_{d_x})^T \succeq 0$, where $|B| \in \mathbb{R}^{n \times m}$ is the signless incidence matrix of G. Let $\tilde{B} \in \mathbb{R}^{(n-1) \times m}$ and $\tilde{R} \in \mathbb{R}^{md_x \times (n-1)d_x}$ be the matrices resulting from removing the row corresponding to sensor 1 from B. Similarly, let $\tilde{D}, \tilde{\mathsf{A}}, \tilde{L}, |\tilde{L}| \in \mathbb{R}^{(n-1)d_x \times (n-1)d_x}$ denote the generalized degree, adjacency, Laplacian, and signless Laplacian matrices with the row and column corresponding to sensor 1 removed. Thm. 2.2.1 in Barooah (2007) shows that $\tilde{L} \succ 0$ provided that G is connected. The same approach can be used to show that $|\tilde{L}| \succ 0$. Let $\tilde{x} \in \mathbb{R}^{(n-1)d_x}$ be the locations of sensors $2, \ldots, n$ in the reference frame of sensor 1 and $\hat{x}_t \in \mathbb{R}^{(n-1)d_x}$ be their estimates at time t obtained from (2.27). The update in (2.27) can be written in matrix form as follows:

$$\tilde{D}\hat{x}_{t+1} = \tilde{\mathsf{A}}\hat{x}_t + \tilde{R}^T \mathcal{E}^{-1} \left(\tilde{R}\tilde{x} + \frac{1}{t+1} \sum_{\tau=0}^{t} \epsilon_\tau \right). \tag{D.15}$$

Define the estimation error at time t as $e_t := \tilde{x} - \hat{x}_t$ and let $u_t := \frac{1}{t+1} \sum_{\tau=0}^{t} \epsilon_\tau$. The dynamics of the error state can be obtained from (D.15):

$$\begin{aligned} e_{t+1} &= \tilde{x} - \tilde{D}^{-1}\tilde{\mathsf{A}}\hat{x}_t - \tilde{D}^{-1}\tilde{L}\tilde{x} - \tilde{D}^{-1}\tilde{R}^T\mathcal{E}^{-1}u_t \\ &= \tilde{x} - \tilde{D}^{-1}\tilde{\mathsf{A}}\hat{x}_t - \tilde{D}^{-1}\left(\tilde{D} - \tilde{A}\right)\tilde{x} - \tilde{D}^{-1}\tilde{R}^T\mathcal{E}^{-1}u_t \\ &= \tilde{D}^{-1}\tilde{\mathsf{A}}e_t - \tilde{D}^{-1}\tilde{R}^T\mathcal{E}^{-1}u_t. \end{aligned}$$

The error dynamics are governed by a stochastic linear time-invariant system, whose internal stability depends on the eigenvalues of $\tilde{D}^{-1}\tilde{\mathsf{A}}$. To show that the error dynamics are stable, we use the following lemma.

Lemma D.8 (Elsner and Mehrmann 1991, Lemma 4.2)**.** *Let $L = D - \mathsf{A} \in \mathbb{C}^{n \times n}$ be such that $D + D^* \succ 0$ and $L_\theta = D + D^* - (e^{i\theta}\mathsf{A} + e^{-i\theta}A^*) \succ 0$ for all $\theta \in \mathbb{R}$. Then $\rho(D^{-1}A) < 1$.*

Consider $\tilde{L}_\theta := 2(\tilde{D} - \cos(\theta)\tilde{\mathsf{A}})$. If $\cos\theta = 0$, then $\tilde{L}_\theta = 2\tilde{D} \succ 0$. If $\cos\theta \in (0,1]$, then $\tilde{L}_\theta \succeq 2\cos\theta\tilde{L} \succ 0$. Finally, if $\cos\theta \in [-1,0)$, then $\tilde{L}_\theta \succeq 2|\cos\theta||\tilde{L}| \succ 0$. Therefore, $\rho\big(\tilde{D}^{-1}\tilde{\mathsf{A}}\big) < 1$. The proof of the theorem is completed by the following lemma with $\mathsf{F} := \tilde{D}^{-1}\tilde{\mathsf{A}}$ and $\mathsf{G} := -\tilde{D}^{-1}\tilde{R}^T\mathcal{E}^{-1}$.

Lemma D.9. *Consider the discrete-time stochastic linear time-invariant system, $e_{t+1} = \mathsf{F}e_t + \mathsf{G}\frac{1}{t+1}\sum_{\tau=0}^{t}\epsilon_\tau$, driven by Gaussian noise $\epsilon_\tau \sim \mathcal{N}(0, \mathcal{E})$, which is independent at any pair of times. If the spectral radius of F satisfies $\rho(\mathsf{F}) < 1$, then $e_t \xrightarrow{a.s., L^2} 0$.*

Proof. By the strong law of large numbers (Durrett 2010, Thm.2.4.1), $u_t := \frac{1}{t+1}\sum_{\tau=0}^{t}\epsilon_\tau$ converges to 0 almost surely. Let Ω be the set with measure 1 on which u_t converges so that for any $\gamma > 0$, $\exists\, T \in \mathbb{N}$ such that $\forall t \geq T$, $\|u_t\| \leq \gamma$. For realizations in Ω, the solution with initial time T is:

$$e_t = \mathsf{F}^{t-T}e_T + \textstyle\sum_{\tau=T}^{t-1}\mathsf{F}^{t-\tau-1}\mathsf{G}u_\tau.$$

Then, $\|e_t\| \leq \|\mathsf{F}^{t-T}e_T\| + \displaystyle\sum_{\tau=T}^{t-1}\left\|\mathsf{F}^{t-\tau-1}\right\|\|\mathsf{G}\|\gamma$. Taking the limit of t and using that F is stable, we have

$$\lim_{t\to\infty}\|e_t\| \leq \left(\sum_{\tau=0}^{\infty}\|\mathsf{F}^\tau\|\right)\|\mathsf{G}\|\gamma.$$

Since $\rho(\mathsf{F}) < 1$, the system is internally (uniformly) exponentially stable, which is equivalent to $\sum_{\tau=0}^{\infty}\|\mathsf{F}^\tau\| \leq \beta$ for some finite constant β (Rugh 1996, Ch.22). Thus, $\lim_{t\to\infty}\|e_t\| \leq \beta\|\mathsf{G}\|\gamma$, which can be made arbitrarily small by choice of γ. We conclude that $e_t \to 0$ on Ω and consequently $e_t \xrightarrow{a.s.} 0$.

Next, we show convergence in L^2. First, consider the propagation of the cross term $C_t := (t+1)\mathbb{E}e_t u_t^T$. Note that $\mathbb{E}[u_t] = 0$ and $\mathbb{E}[u_t u_t^T] = \frac{\mathcal{E}}{t+1}$. Using the fact that ϵ_{t+1} is independent of e_t and u_t we have

$$C_{t+1} = \mathbb{E}\left(\mathsf{F}e_t + \mathsf{G}u_t\right)\left((t+1)u_t + \epsilon_{t+1}\right)^T = \mathsf{F}C_t + (t+1)\mathsf{G}\mathbb{E}[u_t u_t^T] = \mathsf{F}C_t + \mathsf{G}\mathcal{E}.$$

The solution of the above linear time-invariant system is:

$$C_t = \mathsf{F}^t C_0 + \sum_{\tau=0}^{t-1}\mathsf{F}^{t-\tau-1}\mathsf{G}\mathcal{E}$$

and since F is stable: $\lim\limits_{t\to\infty}\mathbb{E}[e_t u_t^T] = \lim\limits_{t\to\infty}\dfrac{1}{t+1}\displaystyle\sum_{\tau=0}^{t-1}\mathsf{F}^\tau\mathsf{G}\mathcal{E} = 0$. Now, consider the second moment of the error:

$$\Sigma_{t+1} := \mathbb{E}[e_{t+1}e_{t+1}^T] = \mathsf{F}\Sigma_t\mathsf{F}^T + \mathsf{F}\left(\mathbb{E}[e_t u_t^T]\right)\mathsf{G}^T + \mathsf{G}\left(\mathbb{E}[u_t e_t^T]\right)\mathsf{F}^T + \frac{1}{t+1}\mathsf{G}\mathcal{E}\mathsf{G}^T = \mathsf{F}\Sigma_t\mathsf{F}^T + Q_t,$$

where $Q_t := \dfrac{1}{t+1}\left(\mathsf{F}C_t\mathsf{G}^T + \mathsf{G}C_t^T\mathsf{F}^T + \mathsf{G}\mathcal{E}\mathsf{G}^T\right)$. As shown above $Q_t \to 0$ as $t \to \infty$, i.e., for any $\delta > 0$, $\exists\, T' \in \mathbb{N}$ such that $\forall t \geq T'$, $\|Q_t\| \leq \delta$. With initial time T', for $t \geq T'$:

$$\Sigma_t = \mathsf{F}^{t-T'}\Sigma_{T'}\left(\mathsf{F}^T\right)^{t-T'} + \sum_{\tau=T'}^{t-1}\mathsf{F}^{t-\tau-1}Q_\tau\left(\mathsf{F}^T\right)^{t-\tau-1}$$

$$\|\Sigma_t\| \leq \left\|\mathsf{F}^{t-T'}\right\|^2\|\Sigma_{T'}\| + \sum_{\tau=0}^{t-T'-1}\|\mathsf{F}^\tau\|^2\delta \leq \alpha^2\mu^{2(t-T')} + \delta\alpha^2\sum_{\tau=0}^{t-T'-1}\mu^{2\tau},$$

where the existence of the constants $\alpha > 0$ and $0 \leq \mu < 1$ is guaranteed by the stability of F. We conclude that $\lim_{t\to\infty} \|\Sigma_t\| \leq \frac{\delta\alpha^2}{1-\mu^2}$, which can be made arbitrarily small by choice of δ. In other words, $e_t \xrightarrow{L^2} 0$. □

D.10 Proof of Theorem 2.12

We use the same notation and approach as in the proof of Thm 2.10, except that now the terms H_i, V_i, M_i, M, ζ_t, ϕ_{it}, C_{it}, B_{it} are time-varying and stochastic because they depend on the location estimates $\hat{x}_{i,t}$. To emphasize this, we denote them by $\widehat{H}_{it}$, $\widehat{V}_{it}$, $\widehat{M}_{it}$, $\widehat{M}_t$, $\widehat{\zeta}_t$, $\widehat{\phi}_{it}$, $\widehat{C}_{it}$, $\widehat{B}_{it}$, where for example $\widehat{M}_{it} := M_i(\hat{x}_{i,t})$. The same linear systems (D.11) describe the evolutions of ω_t and Ω_t except that they are stochastic now and (D.13) becomes:

$$\frac{\omega_{it}}{t+1} = g_{it} + \widehat{C}_{it} y + \widehat{\phi}_{it}, \qquad \widehat{B}_{it} := \frac{\Omega_{it}}{t+1} = G_{it} + \widehat{C}_{it}.$$

We still have that $\mathcal{K}^t \to \pi \mathbf{1}^T$ with $\pi_j > 0$. Also, g_{it}, G_{it}, and b_{it} are still deterministic and converge to zero as $t \to \infty$. The following observations are necessary to conclude that $\widehat{C}_{it}$ still converges to $\sum_{j=1}^n \pi_j M_j$.

Lemma D.10. *If $\hat{x}_{i,t} \xrightarrow{a.s.} x_i$, then $\widehat{M}_{it} \xrightarrow{a.s., L^2} M_i$.*

Proof. Almost sure convergence follows from the continuity of $M_i(\cdot)$ and the continuous mapping theorem (Durrett 2010, Thm.3.2.4). L^2-convergence follows from the boundedness of $M_i(\cdot)$ and the dominated convergence theorem (Durrett 2010, Thm.1.6.7). □

Lemma D.11. *If $a_t \to a$ and $b_t \to b$, then $\frac{1}{t}\sum_{\tau=0}^{t-1} a_{t-\tau} b_\tau \to ab$.*

Proof. The convergence of a_t implies its boundedness, $|a_t| \leq q < \infty$. Then, notice $ab = \frac{1}{t}\sum_{\tau=0}^{t-1} ab$ and

$$\begin{aligned}
\left| \frac{1}{t}\sum_{\tau=0}^{t-1} a_{t-\tau} b_\tau - ab \right| &= \left| \frac{1}{t}\sum_{\tau=0}^{t-1} \big(a_{t-\tau}(b_\tau - b) + (a_{t-\tau} - a) b \big) \right| \\
&\leq \left| \frac{1}{t}\sum_{\tau=0}^{t-1} a_{t-\tau}(b_\tau - b) \right| + \left| \frac{1}{t}\sum_{\tau=0}^{t-1} (a_{t-\tau} - a) b \right| \\
&\leq \left| q \left(\frac{1}{t}\sum_{\tau=0}^{t-1} b_\tau - b \right) \right| + \left| \left(\frac{1}{t}\sum_{\tau=1}^{t} a_\tau - a \right) b \right|,
\end{aligned}$$

where both terms converge to zero since Cesáro means preserve convergent sequences and their limits. □

Combining Lemma D.10, $\left[\mathcal{K}^t\right]_{ij} \to \pi_j$, and Lemma D.11, we have:

$$\frac{1}{t+1}\sum_{\tau=0}^{t-1} \left[\mathcal{K}^{t-\tau-1}\right]_{ij} \widehat{M}_{j\tau} \xrightarrow{a.s.} \left[\pi \mathbf{1}^T\right]_{ij} M_j = \pi_j M_j.$$

Moreover, $0 \le [\mathcal{K}^t]_{ij} \le 1$ and the boundedness of $\widehat{M}_{jt}$ imply, by the bounded convergence theorem (Durrett 2010, Thm.1.6.7), that the sequence above converges in L^2 as well:

$$\widehat{C}_{it} \xrightarrow{a.s.,L^2} \textstyle\sum_{j=1}^n \pi_j M_j \succ 0. \tag{D.16}$$

In turn, (D.16) guarantees that:

$$\widehat{B}_{it}^{-2} = \big(G_{it} + \widehat{C}_{it}\big)^{-2} \xrightarrow{a.s.} \big(\textstyle\sum_{j=1}^n \pi_j M_j\big)^{-2} \tag{D.17}$$

but is not enough to ensure that $\mathbb{E}\big[\widehat{B}_{it}^{-2}\big]$ remains bounded as $t \to \infty$. The parameter $\delta > 0$ is needed to guarantee the boundedness. In particular, define $\widehat{B}_{it}(\delta) := \widehat{B}_{it} + \delta I_{d_y}$. Then

$$\widehat{B}_{it}(\delta)^{-2} = \big(G_{it} + \widehat{C}_{it} + \delta I_{d_y}\big)^{-2} \prec \frac{1}{\delta^2} I_{d_y}$$

and by the bounded convergence theorem and (D.17):

$$\widehat{B}_{it}(\delta)^{-2} \xrightarrow{a.s.,L^1} \big(\textstyle\sum_{j=1}^n \pi_j M_j + \delta I_{d_y}\big)^{-2}, \tag{D.18}$$

so that $\lim_{t\to\infty} \mathbb{E}\big[\widehat{B}_{it}(\delta)^{-2}\big] < \infty$. From (D.16) and the boundedness of $\widehat{B}_{it}(\delta)^{-1}$ and $\widehat{C}_{it}$, we also have:

$$\widehat{B}_{it}(\delta)^{-1}\widehat{C}_{it}\widehat{B}_{it}(\delta)^{-T} \xrightarrow{a.s.,L^2} \left(\sum_{j=1}^n \pi_j M_j + \delta I_{d_y}\right)^{-1} \left(\sum_{j=1}^n \pi_j M_j\right) \left(\sum_{j=1}^n \pi_j M_j + \delta I_{d_y}\right)^{-T} \tag{D.19}$$

Since $\widehat{H}_{it}$ and $\widehat{V}_{it}$ depend solely on $\hat{x}_{i,t}$, they are independent of $v_{i,t}$. Because $\mathbb{E}[v_{j,\tau}] = 0$, $\mathbb{E}[\widehat{H}_{j\tau}^T \widehat{V}_{j\tau}^{-1} v_{j,\tau}] = 0$ and as before $\mathbb{E}[\widehat{\phi}_{it}] = 0$. Since $\widehat{B}_{it}(\delta)$ is independent of $v_{i,t}$ as well, $\mathbb{E}\big[\widehat{B}_{it}(\delta)^{-2}\widehat{\phi}_{it}\big] = 0$ and a result equivalent to (D.14) holds:

$$\begin{aligned} \mathbb{E}[\widehat{B}_{it}(\delta)^{-1}\widehat{\phi}_{it}\widehat{\phi}_{it}^T\widehat{B}_{it}(\delta)^{-T}] &= \mathbb{E}\left[\widehat{B}_{it}(\delta)^{-1}\left(\frac{1}{(t+1)^2}\sum_{j=1}^n\sum_{\tau=0}^{t-1}\big[\mathcal{K}^{t-\tau-1}\big]_{ij}^2 \widehat{M}_{j\tau}\right)\widehat{B}_{it}(\delta)^{-T}\right] \\ &\preceq \frac{1}{t+1}\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-1}\widehat{C}_{it}\widehat{B}_{it}(\delta)^{-T}\right]. \end{aligned} \tag{D.20}$$

Finally, consider the mean squared error:

$$\begin{aligned}
\mathbb{E}\left[\|\hat{y}_{i,t} - y\|_2^2\right] &= \mathbb{E}\left[\left\|\widehat{B}_{it}(\delta)^{-1}\frac{\omega_{it}}{t+1} - \widehat{B}_{it}(\delta)^{-1}\widehat{B}_{it}(\delta)y\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\widehat{B}_{it}(\delta)^{-1}\left(g_{it} + \widehat{C}_{it}y + \widehat{\phi}_{it} - (G_{it} + \widehat{C}_{it} + \delta I_{d_y})y\right)\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\widehat{B}_{it}(\delta)^{-1}(b_{it} + \widehat{\phi}_{it} + \delta y)\right\|_2^2\right] \\
&= \mathbb{E}\Big[b_{it}^T\widehat{B}_{it}(\delta)^{-2}b_{it} + \widehat{\phi}_{it}^T\widehat{B}_{it}(\delta)^{-2}\widehat{\phi}_{it} + \delta^2 y^T\widehat{B}_{it}(\delta)^{-2}y \\
&\qquad + 2b_{it}^T\widehat{B}_{it}(\delta)^{-2}\widehat{\phi}_{it} + 2\delta y^T\widehat{B}_{it}(\delta)^{-2}\widehat{\phi}_{it} + 2\delta b_{it}^T\widehat{B}_{it}(\delta)^{-2}y\Big] \\
&= b_{it}^T\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right]b_{it} + \operatorname{tr}\left(\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-1}\widehat{\phi}_{it}\widehat{\phi}_{it}^T\widehat{B}_{it}(\delta)^{-T}\right]\right) \\
&\qquad + \delta^2 y^T\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right]y + 2\delta b_{it}^T\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right]y \\
&\overset{(D.20)}{\leq} b_{it}^T\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right]b_{it} + 2\delta b_{it}^T\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right]y + \delta^2 y^T\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right]y \\
&\qquad + \frac{1}{t+1}\operatorname{tr}\left(\mathbb{E}\left[\widehat{B}_{it}(\delta)^{-1}\widehat{C}_{it}\widehat{B}_{it}(\delta)^{-T}\right]\right) \\
&\to \delta^2 y^T\left(\sum_{j=1}^n \pi_j M_j + \delta I_{d_y}\right)^{-2}y.
\end{aligned}$$

In the final step, the first two terms go to zero because $b_{it} \to 0$ and $\lim_t \mathbb{E}\left[\widehat{B}_{it}(\delta)^{-2}\right] < \infty$ from (D.18), the third term converges in view of (D.18) again, while the last term goes to zero because the trace is bounded in the limit in view of (D.19). □

D.11 Validity of the Data Association Probability Density Functions

For simplicity, let $p_d(y_i, x) = p_d(y_j, x) \equiv p_d$ for all i, j in this section. We verify that $p(\pi \mid Y_d(x), x)$ is a valid probability density function in each of the following cases.

No missed detections and no false positives

In this case, $\pi \in \Pi_{n,n}$ with likelihood $p(\pi \mid Y_d(x), x) = 1/n!$, which sums to one as follows:

$$\sum_{\pi \in \Pi_{n,n}} p(\pi \mid Y_d(x), x) = \frac{1}{n!}|\Pi_{n,n}| = 1.$$

No false positives but missed detections are possible

In this case, $\pi \in \Pi_{m,n}$ with likelihood:

$$p(\pi \mid Y_d(x), x) = \frac{1}{m!}p_d^m(1 - p_d)^{n-m},$$

which sums to one using the Binomial theorem:

$$\sum_{m=0}^{n}\sum_{\pi\in\Pi_{m,n}} p(\pi \mid Y_d(x),x) = \sum_{m=0}^{n} {}_nP_m \frac{1}{m!} p_d^m (1-p_d)^{n-m} = (p_d + (1-p_d))^n = 1.$$

No missed detections but false positives are possible

In this case, $\pi \in \Pi_{n,m}$ with likelihood:

$$p(\pi \mid Y_d(x),x) = \frac{1}{{}_mP_n} \frac{e^{-\lambda}\lambda^{m-n}}{(m-n)!},$$

which sums to one as follows:

$$\sum_{m=n}^{\infty}\sum_{\pi\in\Pi_{n,m}} p(\pi \mid Y_d(x),x) = \sum_{m=n}^{\infty} {}_mP_n \frac{1}{{}_mP_n} \frac{e^{-\lambda}\lambda^{m-n}}{(m-n)!} = 1.$$

Both missed detections and false positives are possible

The likelihood of $\pi \in \bar{\Pi}_{n,m}$ with k true positive assignments is:

$$p(\pi \mid Y_d(x),x) = \frac{1}{{}_mP_k} p_d^k (1-p_d)^{n-k} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!},$$

which sums to one as follows:

$$\begin{aligned}
\sum_{m=0}^{\infty}\sum_{k=0}^{\min\{m,n\}} \binom{n}{k} {}_mP_k \; p(\pi \mid Y_d(x),x) &= \sum_{m=0}^{\infty}\sum_{k=0}^{\min\{m,n\}} \binom{n}{k} p_d^k (1-p_d)^{n-k} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!} \\
&= \sum_{m=0}^{n}\sum_{k=0}^{m} \binom{n}{k} p_d^k (1-p_d)^{n-k} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!} + \sum_{m=n+1}^{\infty}\sum_{k=0}^{n} \binom{n}{k} p_d^k (1-p_d)^{n-k} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!} \\
&\overset{\text{switch}}{\underset{\text{index order}}{=\!=\!=\!=}} \sum_{k=0}^{n}\sum_{m=k}^{n} \binom{n}{k} p_d^k (1-p_d)^{n-k} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!} + \sum_{k=0}^{n}\sum_{m=n+1}^{\infty} \binom{n}{k} p_d^k (1-p_d)^{n-k} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!} \\
&= \sum_{k=0}^{n} \binom{n}{k} p_d^k (1-p_d)^{n-k} \left[\sum_{m=k}^{\infty} \frac{e^{-\lambda}\lambda^{m-k}}{(m-k)!}\right] = 1.
\end{aligned}$$

D.12 Proof of Theorem 3.2

Let $V_1 := Y_d(x)$ and $V_2 := Z$ be the vertices of a weighted complete bipartite graph $G := (V_1, V_2, E, w)$, where the weight w_e associated with $e := (i,j) \in E$ is $Q(i,j)$. The functions π in (3.9) specify different data associations between the objects V_1 and the measurements V_2. The introduction of missed detections ('0' in the range of π) means that some detectable objects need not to be assigned to a measurement in Z. As any object could be missed, the data associations π correspond to *matchings* (not necessarily perfect as before) in the graph G. Given a matching π, the associated product term inside the sum in (3.9) corresponds to the weight of π. The sum over all π corresponds to the sum of the weights of all matchings in G. The sum of the weights of all k-matchings (matchings with k edges) can be computed via the k-th subpermanent sum of the adjacency matrix Q of G.

Subpermanent Sum. *Let A be an $n \times m$ non-negative matrix with $n \leq m$ and let $Q_{k,n}$ be the set of all subsets of cardinality k of $1, \ldots, n$. For $\alpha \in Q_{k,n}$ and $\beta \in Q_{k,m}$ let $A[\alpha, \beta] := [A(\alpha_i, \beta_j)]_{i,j=1}^{k}$ be the corresponding k-by-k submatrix of A. Define $per_0(A) := 1$ and*

$$\mathbf{per}_k(A) := \sum_{\alpha \in Q_{k,n}, \beta \in Q_{k,m}} \mathbf{per}(A[\alpha, \beta]), \quad k = 1, \ldots, n \tag{D.21}$$

Then, the sum in (3.9) is equal to the sum over the weights of all k-matchings:

$$\sum_{\pi} \prod_{i | \pi(i) > 0} \frac{p_d(y_i, x) p_z(z_{\pi(i)} \mid y_i, x)}{(1 - p_d(y_i, x)) \lambda p_\kappa(z_{\pi(i)})} = \sum_{k=0}^{|Y_d(x)|} \mathbf{per}_k(Q) \tag{D.22}$$

where the assumption that $|Y_d(x)| \leq m$ is used. The following two lemmas describe a reduction from the problem of summing all subpermanent sums of a rectangular matrix (or matchings in an unbalanced bipartite graph) to the problem of the permanent of a rectangular matrix (or perfect matchings in an unbalanced bipartite graph) and then to the problem of the permanent of a square matrix (or perfect matchings in a balanced bipartite graph).

Lemma D.12. *Let $A_{n,m}$ be an $n \times m$ matrix with $n \leq m$. Then,*

$$\sum_{k=0}^{n} \mathbf{per}_k(A_{n,m}) = \mathbf{per}\left(\begin{bmatrix} A_{n,m} & I_n \end{bmatrix}\right).$$

Proof: Associate A with a weighted complete bipartite graph $G_A := (V_1 := \{1, \ldots, n\}, V_2 := \{1, \ldots, m\}, E, w_A)$, where the weights w_A corresponding with the entries of A. To obtain the graph G_B associated with $B := \begin{bmatrix} A_{n,m} & I_n \end{bmatrix}$ add n dummy nodes V_3 to V_2 and n edges of weight 1. For $k \in \{0, \ldots, n\}$, fix subsets $\alpha \in Q_{k,n}$ and $\beta \in Q_{k,m}$ using the notation from (D.21). A perfect matching in G_B associated with α and β corresponds to:

- A k-matching between $\alpha \in V_1$ and $\beta \in V_2$ of weight $\mathbf{per}(A[\alpha, \beta])$
- A $(n-k)$-matching between $V_1 \setminus \alpha$ and V_3 of weight 1

Then, $\mathbf{per}(B)$ is the sum of all perfect matchings in G_B:

$$\mathbf{per}(B) = \sum_{k=0}^{n} \sum_{\substack{\beta \in Q_{k,m} \\ \alpha \in Q_{k,n}}} \mathbf{per}(A[\alpha, \beta]) = \sum_{k=0}^{n} \mathbf{per}_k(A),$$

where the last equality follows directly from Def. (D.21). □

Lemma D.13. *Let $A_{n,m}$ be an $n \times m$ matrix with $n \leq m$. Then,*

$$\mathbf{per}(A_{n,m}) = \frac{1}{(m-n)!} \mathbf{per}\left(\begin{bmatrix} A_{n,m} \\ 1_{m-n,m} \end{bmatrix}\right)$$

where $1_{m-n,m}$ is a $(m-n) \times m$ matrix of all ones.

Proof: Associate A with a weighted complete bipartite graph $G_A := (V_1 := \{1,\ldots,n\}, V_2 := \{1,\ldots,m\}, E, w_A)$, where the weights w_A correspond with the entries of A. To obtain the graph G_B associated with $B := \begin{bmatrix} A_{n,m}^T & 1_{m-n,m}^T \end{bmatrix}^T$ add $(m-n)$ dummy nodes V_3 to V_1 and $(m-n)m$ edges of weight 1. Fix a subset $\beta \in Q_{m-n,m}$ using the notation from (D.21). A perfect matching in G_B associated with β corresponds to:

- A n-matching between V_1 and $V_2 \setminus \beta$ of weight $\mathbf{per}(A[V_1, V_2 \setminus \beta])$
- A $(m-n)$-matching between V_3 and β of weight $(m-n)!$

Then, $\mathbf{per}(B)$ is the sum of all perfect matchings in G_B:

$$\begin{aligned}\mathbf{per}(B) &= \sum_{\beta \in Q_{m-n,m}} (m-n)!\,\mathbf{per}(A[V_1, V_2 \setminus \beta]) \\ &= (m-n)!\,\mathbf{per}(A),\end{aligned}$$

where the last equality follows directly from Def. (D.21). □

The proof is completed by combining the two reductions above to write the sum in (D.22) as:

$$\sum_{k=0}^{|Y_d(x)|} \mathbf{per}_k(Q) = \frac{1}{m!}\,\mathbf{per}\left(\begin{bmatrix} Q & I_{|Y_d(x)|} \\ 1_{m,m} & 1_{m,|Y_d(x)|} \end{bmatrix}\right). \qquad \square$$

D.13 Summary of the Semantic Observation Models

Table D.1: *No missed detections and no clutter*: the likelihood $p(Z \mid Y_d(x), x)$ of a set of semantic observations Z is shown for different combinations of $m := |Z|$ and $n := |Y_d(x)|$. The dependence of the likelihoods on x is omitted for clarity.

$n \neq m$	0
$0 = m = n$	1
$0 < m = n$	$\dfrac{1}{m!}\,\mathbf{per}\begin{bmatrix} p_z(z_1 \mid y_1) & \cdots & p_z(z_m \mid y_1) \\ \vdots & & \vdots \\ p_z(z_1 \mid y_n) & \cdots & p_z(z_m \mid y_n) \end{bmatrix}$

Table D.2: *No clutter but missed detections are possible*: the likelihood $p(Z \mid Y_d(x), x)$ of a set of semantic observations Z is shown for different combinations of $m := |Z|$ and $n := |Y_d(x)|$. The dependence of the likelihoods on x is omitted for clarity.

$n < m$	0
$0 = m \leq n$	$\prod_{i=1}^{n} (1 - p_d(y_i))$
$0 < m \leq n$	$\dfrac{1}{m!(n-m)!}\,\mathbf{per}\begin{bmatrix} p_d(y_1)p_z(z_1 \mid y_1) & \cdots & p_d(y_1)p_z(z_m \mid y_1) & 1-p_d(y_1) & \cdots & 1-p_d(y_1) \\ \vdots & & \vdots & \vdots & & \vdots \\ p_d(y_n)p_z(z_1 \mid y_n) & \cdots & p_d(y_n)p_z(z_m \mid y_n) & 1-p_d(y_n) & \cdots & 1-p_d(y_n) \end{bmatrix}$

Table D.3: *No missed detections but clutter is possible*: the likelihood $p(Z \mid Y_d(x), x)$ of a set of semantic observations Z is shown for different combinations of $m := |Z|$ and $n := |Y_d(x)|$. The dependence of the likelihoods on x is omitted for clarity.

$m < n$	0
$0 = n \leq m$	$\frac{e^{-\lambda}\lambda^m}{m!} \prod_{j=1}^{m} p_\kappa(z_j)$
$0 < n \leq m$	$\frac{e^{-\lambda}}{m!(m-n)!}\,\mathbf{per}\begin{bmatrix} p_z(z_1 \mid y_1) & \cdots & p_z(z_m \mid y_1) \\ \vdots & & \vdots \\ p_z(z_1 \mid y_n) & \cdots & p_z(z_m \mid y_n) \\ \lambda p_\kappa(z_1) & \cdots & \lambda p_\kappa(z_m) \\ \vdots & & \vdots \\ \lambda p_\kappa(z_1) & \cdots & \lambda p_\kappa(z_m) \end{bmatrix}$

Table D.4: *Both missed detections and clutter are possible*: the likelihood $p(Z \mid Y_d(x), x)$ of a set of semantic observations Z is shown for different combinations of $m := |Z|$ and $n := |Y_d(x)|$. The dependence of the likelihoods on x is omitted for clarity.

$n = 0$	$\frac{e^{-\lambda}\lambda^m}{m!} \prod_{j=1}^{m} p_\kappa(z_j)$
$m = 0$	$\prod_{i=1}^{n} (1 - p_d(y_i))$
$0 < n \leq m$	$\frac{e^{-\lambda}}{m!m!}\,\mathbf{per}\begin{bmatrix} \tfrac{p_d(y_1)p_z(z_1 \mid y_1)}{\lambda p_\kappa(z_1)} & \cdots & \tfrac{p_d(y_1)p_z(z_m \mid y_1)}{\lambda p_\kappa(z_m)} & 1 - p_d(y_1) & & 0 \\ \vdots & & \vdots & & \ddots & \\ \tfrac{p_d(y_n)p_z(z_1 \mid y_n)}{\lambda p_\kappa(z_1)} & \cdots & \tfrac{p_d(y_n)p_z(z_m \mid y_n)}{\lambda p_\kappa(z_m)} & 0 & & 1 - p_d(y_n) \\ \lambda p_\kappa(z_1) & \cdots & \cdots & \cdots & \cdots & \lambda p_\kappa(z_1) \\ \vdots & & & & & \vdots \\ \lambda p_\kappa(z_m) & \cdots & \cdots & \cdots & \cdots & \lambda p_\kappa(z_m) \end{bmatrix}$
$0 < m \leq n$	$\frac{e^{-\lambda}}{m!n!}\,\mathbf{per}\begin{bmatrix} \tfrac{p_d(y_1)p_z(z_1 \mid y_1)}{\lambda p_\kappa(z_1)} & \cdots & \tfrac{p_d(y_1)p_z(z_m \mid y_1)}{\lambda p_\kappa(z_m)} & 1 - p_d(y_1) & \cdots & 1 - p_d(y_1) \\ \vdots & & \vdots & \vdots & & \vdots \\ \tfrac{p_d(y_n)p_z(z_1 \mid y_n)}{\lambda p_\kappa(z_1)} & \cdots & \tfrac{p_d(y_n)p_z(z_m \mid y_n)}{\lambda p_\kappa(z_m)} & 1 - p_d(y_n) & \cdots & 1 - p_d(y_n) \\ \lambda p_\kappa(z_1) & & 0 & \lambda p_\kappa(z_1) & \cdots & \lambda p_\kappa(z_1) \\ & \ddots & & \vdots & & \vdots \\ 0 & & \lambda p_\kappa(z_m) & \lambda p_\kappa(z_m) & \cdots & \lambda p_\kappa(z_m) \end{bmatrix}$

D.14 Active Bearing-only Localization

Bearing-only entropy minimization solves the active semantic localization problem in (3.12) but the pose entropy is conditioned only on the future bearing measurements:

$$\begin{aligned} \sigma^* \in \operatorname*{arg\,min}_{\sigma \in \Sigma} \;& \mathbf{H}(x_{0:T} \mid B_{1:T}) \\ \text{s.t.} \;& x_{s+1} = f(x_s, \sigma_s, v_s), \;\; s = 0, \ldots, T-1 \\ & B_s = \{\beta(x_s, y) + \eta_s \mid y^p \in FoV(x_s)\}, \; s = 1, \ldots, T, \end{aligned} \tag{D.23}$$

where $\eta_s \sim \mathcal{N}(0, \Sigma_\beta)$ is the bearing-measurement noise and B_s is the set of bearing measurements obtained at time s. Since the robot motion model $f(x, u, v)$ and the bearing measurement model $\beta(x, y)$ are continuous functions of the robot state x and landmark states y, perturbed by Gaussian noise, we can linearize them as discussed in Sec. 2.4.6. In detail, let $\bar{x}_s$ be the mean of the particle distribution at time s and define $\delta x_s := x_s - \bar{x}_s$. We linearize the constraints in (D.23) around the means $\bar{x}_s$ and $\beta(\bar{x}_s, y)$ to obtain:

$$\begin{aligned} \sigma^* \in \operatorname*{arg\,min}_{\sigma \in \Sigma} \;& \mathbf{H}(\delta x_{0:T}) - \mathbf{I}(\delta x_{0:T}; \delta B_{1:T}) \\ \text{s.t.} \;& \bar{x}_{s+1} = f(\bar{x}_s, \sigma_s, 0), \;\; s = 0, \ldots, T-1 \\ & \delta x_s \approx \left[\frac{\partial f}{\partial x}(\bar{x}_s, \sigma_s, 0)\right] \delta x_s + \left[\frac{\partial f}{\partial v}(\bar{x}_s, \sigma_s, 0)\right] v_s \\ & \delta B_s \approx \left\{ \frac{\partial \beta}{\partial x}(\bar{x}_s, y) \delta x_s + \eta_s \;\middle|\; y^p \in FoV(x_s) \right\}, \; s = 1, \ldots, T. \end{aligned} \tag{D.24}$$

Let C_0 be the covariance of the prior particle distribution and assume that $\delta x_0 \sim \mathcal{N}(0, C_0)$. Since the constraints in (D.24) are linear and the measurement noise is Gaussian, the distribution of δx_s remains Gaussian for $s = 1, \ldots, T$. In particular, it can be computed via the Kalman filter. In addition, the entropy and mutual information of Gaussian random variables depend only on the associated covariance matrices (and not on the particular measurement realization $\delta B_{1:T}$) and can be computed in closed form. Thus, we compute the cost for each control sequence $\sigma \in \Sigma$ and choose the sequence with the lowest cost. Refer to Sec. 2.4.6 for more details regarding conditional entropy minimization via linearization and model predictive control.

D.15 Proof of Theorem 3.7

From Thm. 2.10, $\hat{x}^i(k) \overset{L^2}{\to} x^*$, $\forall i$, which implies convergence in L^1 and in probability. Convergence in L^1 implies that the sequence $\{\hat{x}^i(k)\}$ is uniformly integrable (UI) for all i (Durrett 2010, Thm. 5.5.2). We claim that this implies that the sequence of finite-difference weights $W(\hat{x}^i(k))$ computed in (3.24) is UI for each i. The matrix Φ in (3.24) is a bounded continuous function of $\hat{x}^i(k)$, which means that there exists a constant $K_i^\Phi \leq \infty$ for each i

such that $\|\Phi(\hat{x}^i(k))^{-T}\|_1 \leq K_i^{\Phi}$. Define $\alpha_i(k) := \hat{x}_i^i(k) - \sum_{j=1}^n \hat{x}_j^i(k)/n$. From (3.25):

$$\begin{aligned}
\|W(\hat{x}^i(k))\|_1 &\leq \left\| \begin{bmatrix} 2\delta^2 e^{-\delta^2\|\alpha_1(k)\|_2^2}\alpha_1^T(k) \\ \vdots \\ 2\delta^2 e^{-\delta^2\|\alpha_n(k)\|_2^2}\alpha_n^T(k) \end{bmatrix}^T \right\|_1 \left\|\Phi(\hat{x}^i(k))^{-T}\right\|_1 \\
&\leq 2\delta^2 K_i^{\Phi} \sum_{j=1}^n e^{-\delta^2\|\alpha_j(k)\|_2^2}\|\alpha_j(k)\|_1 \\
&\leq 2\delta^2 K_i^{\Phi} \sum_{j=1}^n \left\| \hat{x}_j^i(k) - \frac{1}{n}\sum_{l=1}^n \hat{x}_l^i(k) \right\|_1 \\
&\leq 4\delta^2 K_i^{\Phi} \sum_{j=1}^n \|\hat{x}_j^i(k)\|_1 = 4\delta^2 K_i^{\Phi}\|\hat{x}^i(k)\|_1.
\end{aligned}$$

By UI of $\{\hat{x}^i(k)\}$, for any $\epsilon > 0$, there exist $K_i \in [0,\infty)$ such that $\mathbb{E}\big[\|\hat{x}^i(k)\|_1 \mathbb{1}_{\{\|\hat{x}^i(k)\|_1 \geq K_i\}}\big] \leq \epsilon$ for all k. Then for all i, k:

$$\begin{aligned}
\mathbb{E}\big[\|W(\hat{x}^i(k))\|_1 \mathbb{1}_{\{\|W(\hat{x}^i(k))\|_1 \geq 4\delta^2 K_i^{\Phi} K_i\}}\big] &\leq 4\delta^2 K_i^{\Phi}\mathbb{E}\Big[\|\hat{x}^i(k)\|_1 \mathbb{1}_{\{4\delta^2 K_i^{\Phi}\|\hat{x}^i(k)\|_1 \geq 4\delta^2 K_i^{\Phi} K_i\}}\Big] \\
&\leq 4\delta^2 K_i^{\Phi}\epsilon.
\end{aligned}$$

Since $W(\hat{x}^i(k))$ is a continous function of $\hat{x}_i(k)$ by the continuous mapping theorem, $W(\hat{x}^i(k)) \xrightarrow{p} W(x^*)$, $\forall i$. This, coupled with the uniform integrability of $\{W(\hat{x}^i(k))\}$ for all i implies that $W(\hat{x}^i(k)) \xrightarrow{L^1} W(x^*)$, $\forall i$. The signal measurements $z_i(\tau)$ in (3.36) are independent of the estimates $W(\hat{x}^i(k))$ because the latter are based on the relative measurements in (3.33). Therefore,

$$\begin{aligned}
\mathbb{E}\hat{g}_i(k) &= \mathbb{E}\big[\mathbf{col}_i(W(\hat{x}^i(k)))\big]\tfrac{1}{k+1}\textstyle\sum_{\tau=0}^k \mathbb{E}z_i(\tau) \\
&= \mathbb{E}\big[\mathbf{col}_i(W(\hat{x}^i(k)))\big]h(x_i^*, y) \to \mathbf{col}_i(W(x^*))h(x_i^*, y).
\end{aligned} \tag{D.25}$$

Now, consider the behavior of the consensus filter in (3.37) with $\mu_{i,k} = \hat{g}_i(k)$. Eliminating the state $q_{i,k}$ and writing the equations in matrix form gives:

$$r_{k+1} = (I_{nd_x} - \beta(L \otimes I_{d_x}))\, r_k + (\mu_{k+1} - \mu_k),$$

where L is the Laplacian of the communication graph G. Taking expectations above results is a deterministic linear time-invariant system, which was analyzed in Spanos et al. (2005). In light of (D.25), Proposition 1 in Spanos et al. (2005) shows that for all i:

$$\lim_{k\to\infty}\left(\mathbb{E}[r_{i,k}] - \frac{1}{n}\sum_{i=1}^n \mathbf{col}_i(W(x^*))h(x_i^*, y)\right) = 0.$$

Finally, the finite-difference gradient approximation in (3.22) shows that:

$$\lim_{k\to\infty}\mathbb{E}[r_{ik}] = \frac{1}{n}W(x^*)\begin{pmatrix} h(x_1^*, y) \\ \vdots \\ h(x_n^*, y)\end{pmatrix} = \frac{1}{n}\Big(g(m^*, y) + b\Big),\ \forall i.$$

D.16 Multimedia Extensions

Table D.5: Index to Multimedia Extensions

Extension	Media Type	Description
1	Video	Multi-robot active exploration, localization, and mapping
2	Video	Mobile robot localization from semantic observations
3	Video	Global positioning of the Tango phone
4	Video	Global positioning of the Tango phone
5	Video	Global semantic localization on KITTI dataset sequence 00 with several restarts
6	Video	Global semantic localization on KITTI dataset sequence 00
7	Video	Global semantic localization on KITTI dataset sequence 05
8	Video	Global semantic localization on KITTI dataset sequence 06
9	Video	Global semantic localization on KITTI dataset sequence 07
10	Video	Global semantic localization on KITTI dataset sequence 08
11	Video	Global semantic localization on KITTI dataset sequence 09
12	Video	Global semantic localization on KITTI dataset sequence 10
13	Video	Global semantic localization on KITTI dataset sequence 08 (fail case)
14	Data	Car and window positions used for the semantic maps in the KITTI dataset experiments
15	Video	Wireless radio source seeking with a single robot using random-direction stochastic gradient ascent
16	Video	Active object recognition with a depth camera attached to the wrist of a PR2 robot
17	Video	Single-image object recognition via active deformable part models on the Pascal visual object classes 2007 dataset

Bibliography

R. Anati, D. Scaramuzza, K. Derpanis, and K. Daniilidis. Robot Localization Using Soft Object Detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4992–4999, 2012.

A. Angeli, S. Doncieux, J. Meyer, and D. Filliat. Visual topological SLAM and global localization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4300–4305, 2009.

J. Aspnes, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, Y. Yang, B. Anderson, and P. Belhumeur. A Theory of Network Localization. *IEEE Trans. on Mobile Computing*, 5(12):1663–1678, 2006.

N. Atanasov, J. Le Ny, N. Michael, and G. Pappas. Stochastic Source Seeking in Complex Environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3013–3018, 2012.

N. Atanasov, B. Sankaran, J. Le Ny, T. Koletschka, G. Pappas, and K. Daniilidis. Hypothesis Testing Framework for Active Object Detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4216–4222, 2013.

N. Atanasov, J. Le Ny, K. Daniilidis, and G. Pappas. Information Acquisition with Sensing Robots: Algorithms and Error Bounds. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 6447–6454, 2014a.

N. Atanasov, B. Sankaran, J. Le Ny, G. Pappas, and K. Daniilidis. Nonmyopic View Planning for Active Object Classification and Pose Estimation. *IEEE Trans. on Robotics (TRO)*, 30(5): 1078–1090, 2014b.

N. Atanasov, R. Tron, V. Preciado, and G. Pappas. Joint Estimation and Localization in Sensor Networks. In *IEEE Conf. on Decision and Control (CDC)*, pages 6875–6882, 2014c.

N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Semantic Localization Via the Matrix Permanent. In *Robotics: Science and Systems (RSS)*, 2014d.

N. Atanasov, J. Le Ny, K. Daniilidis, and G. Pappas. Decentralized Active Information Acquisition: Theory and Application to Multi-Robot SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4775–4782, 2015a.

N. Atanasov, J. Le Ny, and G. Pappas. Distributed Algorithms for Stochastic Source Seeking with Mobile Robot Networks. *ASME Journal of Dynamic Systems, Measurement, and Control (JDSMC)*, 137(3), 2015b.

N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Localization from Semantic Observations via the Matrix Permanent. *International Journal of Robotics Research*, 2015c.

M. Athans. On the Determination of Optimal Costly Measurement Strategies for Linear Stochastic Systems. *Automatica*, 8(4):397–412, 1972.

A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the Real World: Active Visual Object Search Based on Spatial Relations. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2818–2824, 2011.

S. Azuma, M. Sakar, and G. Pappas. Stochastic Source Seeking by Mobile Robots. *IEEE Trans. on Automatic Control*, 57(9):2308–2321, 2012.

T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, The University of Sydney, 2002.

R. Bajcsy. Active Perception. *Proc. of the IEEE*, 76(8), 1988.

S. Bao and S. Savarese. Semantic Structure from Motion. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2025–2032, 2011.

Y. Bar-Shalom, F. Daum, and J. Huang. The Probabilistic Data Association Filter. *IEEE Control Systems*, 29(6):82–100, 2009.

T. Barfoot, C. Tong, and S. Sarkka. Batch Continuous-Time Trajectory Estimation as Exactly Sparse Gaussian Process Regression. In *Robotics: Science and Systems (RSS)*, 2014.

A. Barker, D. Brown, and W. Martin. Bayesian estimation and the Kalman filter. *Computers and Mathematics with Applications*, 30(10), 1995.

P. Barooah and J. Hespanha. Estimation on Graphs from Relative Measurements. *IEEE Control Systems*, 27(4), 2007.

P. Barooah. *Estimation and Control with Relative Measurements: Algorithms and Scaling Laws*. PhD thesis, University of California Santa Barbara, 2007.

D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

I. Bezáková, D. Štefankovič, V. Vazirani, and E. Vigoda. Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 900–907, 2006.

G. Bierman. Sequential Square Root Filtering and Smoothing of Discrete Linear Systems. *Automatica*, 10(2):147–158, 1974.

A. Bishop and P. Jensfelt. Global Robot Localization with Random Finite Set Statistics. In *Int. Conf. on Information Fusion*, 2010.

V. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. Appearance-based Active Object Recognition. *Image and Vision Computing*, 18(9):715–727, 2000.

L. Bourdev and J. Brandt. Robust Object Detection via Soft Cascade. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 236–243, 2005.

F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte. Information Based Adaptive Robotic Exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 540–545, 2002.

L. Brinón-Arranz and L. Schenato. Consensus-based Source-seeking with a Circular Formation of Agents. In *European Control Conf.*, 2013.

B. Browatzki, V. Tikhanoff, G. Metta, H. Bulthoff, and C. Wallraven. Active Object Recognition on a Humanoid Robot. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2021–2028, 2012.

C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A Survey of Monte Carlo Tree Search Methods. *IEEE Trans on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

S. Brubaker, J. Wu, J. Sun, M. Mullin, and J. Rehg. On the Design of Cascades of Boosted Ensembles for Face Detection. *International Journal of Computer Vision (IJCV)*, 77(1-3):65–86, 2008.

F. Capulli, C. Monti, M. Vari, and F. Mazzenga. Path Loss Models for IEEE 802.11a Wireless Local Area Networks. *Sym. Wireless Comm. Sys.*, 2006.

L. Carlone, J. Du, M. Kaouk Ng, B. Bona, and M. Indri. Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence. *Intelligent & Robotic Systems*, 75(2):291–311, 2014.

H. Carrillo, P. Dames, V. Kumar, and J. Castellanos. Autonomous Robotic Exploration Using Occupancy Grid Maps and Graph SLAM Based on Shannon and Rényi Entropy. In *IEEE Int. Conf. on Robotics and Automation*, 2015a.

H. Carrillo, Y. Latif, M. Rodriguez-Arevalo, J. Neira, and J. Castellanos. On the Monotonicity of Optimality Criteria during Exploration in Active SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015b.

B. Charrow, S. Liu, V. Kumar, and N. Michael. Information-Theoretic Mapping Using Cauchy-Schwarz Quadratic Mutual Information. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

B. Charrow, V. Kumar, and N. Michael. Approximate Representations for Multi-Robot Control Policies that Maximize Mutual Information. In *Robotics: Science and Systems (RSS)*, 2013.

H. Choi. *Adaptive Sampling and Forecasts With Mobile Sensor Networks*. PhD thesis, Massachusetts Institute of Technology, February 2009.

J. Choi, S. Oh, and R. Horowitz. Distributed Learning and Cooperative Control for Multi-agent Systems. *Automatica*, 45(12):2802–2814, 2009.

T. Chung, J. Burdick, and R. Murray. A Decentralized Motion Coordination Strategy for Dynamic Target Tracking. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2416–2422, 2006.

J. Civera, D. Galvez-Lopez, L. Riazuelo, J. Tardos, and J. Montiel. Towards Semantic SLAM Using a Monocular Camera. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1277–1284, 2011.

J. Collins and J. Uhlmann. Efficient Gating in Data Association with Multivariate Gaussian Distributed States. *IEEE Trans. on Aerospace and Electronic Systems*, 28(3):909–916, 1992.

J. Cortés. Distributed Kriged Kalman filter for spatial estimation. *Trans. on Automatic Control (TAC)*, 54(12), 2009.

T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *International Journal of Robotics Research (IJRR)*, 27(6):647–665, 2008.

A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent Distributed Smoothing and Mapping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5220–5227, 2013.

A. Dame, V. Prisacariu, C. Ren, and I. Reid. Dense Reconstruction Using 3D Object Shape Priors. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1288–1295, 2013.

P. Dames and V. Kumar. Cooperative Multi-Target Localization with Noisy Sensors. In *IEEE Int. Conf. on Robotics and Automation*, 2013.

P. Dames, M. Schwager, V. Kumar, and D. Rus. A Decentralized Control Policy for Adaptive Information Gathering in Hazardous Environments. In *IEEE Conf. on Decision and Control (CDC)*, 2012.

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 and Automation Magazine*, 21(2): 46–52, 2013.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research (IJRR)*, 25(12):1181–1203, 2006.

J. Denzler and C. Brown. Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24 (2):145–157, 2002.

J. Derenick and J. Spletzer. Convex Optimization Strategies for Coordinating Large-Scale Robot Formations. *IEEE Trans. on Robotics*, 23(6):1252–1259, 2007.

Y. Diao, Z. Lin, M. Fu, and H. Zhang. A New Distributed Localization Method for Sensor Networks. In *Asian Control Conf. (ASCC)*, 2013.

P. Dollár, R. Appel, and W. Kienzle. Crosstalk Cascades for Frame-Rate Pedestrian Detection. In *European Conference on Computer Vision (ECCV)*, pages 645–659. Springer, 2012.

R. Durrett. *Probability: Theory and Examples*. Cambridge University Press, 4 edition, 2010.

R. Eidenberger and J. Scharinger. Active Perception and Scene Modeling by Planning with Probabilistic 6D Object Poses. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

L. Elsner and V. Mehrmann. Convergence of block iterative methods for linear systems arising in the numerical solution of Euler equations. *Numerische Mathematik*, 59(1), 1991.

M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade Object Detection with Deformable Part Models. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2241–2248, 2010a.

P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645, 2010b.

F. Fleuret and D. Geman. Coarse-to-Fine Face Detection. *International Journal of Computer Vision (IJCV)*, 41(1-2):85–107, 2001.

B. Fornberg, E. Lehto, and C. Powell. Stable Calculation of Gaussian-based RBF-FD Stencils. *Computers & Math. with Applications*, 65(4):627–637, 2013.

C. Forster, M. Pizzoli, and D. Scaramuzza. Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles. In *Robotics: Science and Systems (RSS)*, 2014.

D. Fox, W. Burgard, and S. Thrun. Active Markov Localization for Mobile Robots. *Robotics and Autonomous Systems*, 25(3-4):195–207, 1998.

C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical Semantic Maps for Mobile Robotics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2278–2283, 2005.

T. Gao and D. Koller. Active Classification based on Value of Classifier. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1062–1070, 2011.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013.

A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3d Reconstruction in Real-time. In *Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

S. Gelly and D. Silver. Combining Online and Offline Knowledge in UCT. In *Int. Conf. on Machine Learning*, pages 273–280, 2007.

N. Ghods and M. Kristić. Source Seeking With Very Slow or Drifting Sensors. *Journal of Dynamic Systems, Measurement, and Control*, 133(4), 2011.

M. Göbelbecker, C. Gretton, and R. Dearden. A Switching Planner for Combined Task and Observation Planning. In *AAAI Conference on Artificial Intelligence*, 2011.

A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

Google ATAP group. Project Tango. `https://www.google.com/atap/projecttango`, 2014.

N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.

B. Grocholsky. *Information Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sidney, 2002.

G. Gualdi, A. Prati, and R. Cucchiara. Multistage Particle Windows for Fast and Accurate Object Detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 34(8):1589–1604, 2012.

V. Gupta, T. Chung, B. Hassibi, and R. Murray. On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Sensor Coverage. *Automatica*, 42(2):251–260, 2006.

M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender. Exploiting Probabilistic Knowledge under Uncertain Sensing for Efficient Robot Behaviour. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2011.

K. Hauser. Randomized Belief-Space Replanning in Partially-Observable Continuous Spaces. In *Algorithmic Foundations of Robotics IX*, volume 68 of *Springer Tracts in Advanced Robotics*, pages 193–209. Springer, 2011.

V. Hernandez Bennetts, A. Lilienthal, A. Khaliq, V. Pomareda Sese, and M. Trincavelli. Towards Real-World Gas Distribution Mapping and Leak Localization Using a Mobile Robot with 3D and Remote Gas Sensing Capabilities. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2335–2340, 2013.

A. Hero III and D. Cochran. Sensor Management: Past, Present, and Future. *IEEE Sensors Journal*, 11(12), 2011.

J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. Towards Consistent Vision-Aided Inertial Navigation. In *Algorithmic Foundations of Robotics X*, volume 86 of *Springer Tracts in Advanced Robotics*. 2013.

G. Hoffmann and C. Tomlin. Mobile Sensor Network Control Using Mutual Information Methods and Particle Filters. *IEEE Trans. on Automatic Control*, 55(1):32–47, 2010.

G. Hollinger and G. Sukhatme. Sampling-based Motion Planning for Robotic Information Gathering. In *Robotics: Science and Systems (RSS)*, 2013.

G. Hollinger and G. Sukhatme. Sampling-based Robotic Information Gathering Algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.

G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme. Active Planning for Underwater Inspection and the Benefit of Adaptivity. *The International Journal of Robotics Research*, 32(1): 3–18, 2013a.

G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme. Active Planning for Underwater Inspection and the Benefit of Adaptivity. *The International Journal of Robotics Research*, 32(1): 3–18, 2013b.

A. Howard and B. Gerkey. Adaptive Monte-Carlo Localization (AMCL) package. Robot Operating System (ROS), `http://wiki.ros.org/amcl`, 2002.

M. Huber. *Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications*, volume 19. KIT Scientific Publishing, 2015.

M. Huber. *Probabilistic Framework for Sensor Management*. PhD thesis, Universität Karlsruhe (TH), 2009.

V. Indelman, L. Carlone, and F. Dellaert. Planning Under Uncertainty in the Continuous Domain: a Generalized Belief Space Approach. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

M. Jadaliha, J. Lee, and J. Choi. Adaptive Control of Multiagent Systems for Finding Peaks of Uncertain Static Fields. *Journal of Dynamic Systems, Measurement, and Control*, 134(5), 2012.

M. Jerrum, A. Sinclair, and E. Vigoda. A Polynomial-time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *Journal of the ACM*, 51(4):671–697, 2004.

S. Joshi and S. Boyd. Sensor Selection via Convex Optimization. *IEEE Trans. on Signal Processing*, 57(2):451–462, 2009.

B. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed robotic sensor networks: An information-theoretic approach. *International Journal of Robotics Research (IJRR)*, 31(10):1134–1154, 2012.

L. Kaelbling and T. Lozano-Pérez. Integrated Task and Motion Planning in Belief Space. *The International Journal of Robotics Research*, 32(9-10):1194–1227, 2013.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics (TRO)*, 24(6):1365–1378, 2008.

M. Kaess. *Incremental Smoothing and Mapping*. PhD thesis, Georgia Institute of Technology, 2008.

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. *International Journal of Robotics Research (IJRR)*, 31(2):216–235, 2012.

B. Kalyan, K. Lee, and W. Wijesoma. FISST-SLAM: Finite Set Statistical Approach to Simultaneous Localization and Mapping. *International Journal of Robotics Research (IJRR)*, 29(10):1251–1262, 2010.

K. Kampa, E. Hasanbelliu, and J. Principe. Closed-form Cauchy-Schwarz PDF Divergence for Mixture of Gaussians. In *Int. Joint Conf. on Neural Networks (IJCNN)*, pages 2578–2585, 2011.

A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Gaussian Processes for Object Categorization. *International Journal of Computer Vision (IJCV)*, 88(2):169–188, 2010.

S. Kar, J. Moura, and K. Ramanan. Distributed Parameter Estimation in Sensor Networks: Nonlinear Observation Models and Imperfect Communication. *Trans. on Information Theory*, 58(6), 2012.

S. Karaman and E. Frazzoli. Sampling-based Algorithms for Optimal Motion Planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

V. Karasev, A. Chiuso, and S. Soatto. Controlled Recognition Bounds for Visual Learning and Exploration. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2924–2932, 2012.

S. Karayev, M. Fritz, and T. Darrell. Anytime Recognition of Objects and Scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 572–579, 2014.

U. Khan, S. Kar, and J. Moura. Distributed Sensor Localization in Random Environments Using Minimal Number of Anchor Nodes. *Trans. on Signal Processing*, 57(5), 2009.

U. Khan, S. Kar, A. Jadbabaie, and J. Moura. On connectivity, observability, and stability in distributed estimation. In *Conf. on Decision and Control (CDC)*, 2010.

J. Kiefer. General Equivalence Theory for Optimum Designs (Approximate Theory). *The Annals of Statistics*, 2(5):849–879, 1974.

D. Ko, C. Yi, and I. Suh. Semantic Mapping and Navigation: A Bayesian Approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2630–2636, 2013.

L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo Planning. In *European Conference on Machine Learning (ECML)*, pages 282–293, 2006.

I. Kokkinos. Rapid Deformable Object Detection using Dual-Tree Branch-and-Bound. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2681–2689, 2011.

D. Koller and N. Friedman. *Probabilistic Graphical Models*. MIT press, 2009.

M. Kontitsis, E. Theodorou, and E. Todorov. Multi-Robot Active SLAM with Relative Entropy Optimization. In *American Control Conference (ACC)*, pages 2757–2764, 2013.

J. Košecká and F. Li. Vision Based Topological Markov Localization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1481–1486, 2004.

I. Kostavelis and A. Gasteratos. Learning Spatially Semantic Representations for Cognitive Robot Navigation. *Robotics and Autonomous Systems*, 61(12):1460–1475, 2013.

A. Krause. *Optimizing Sensing*. PhD thesis, Carnegie Mellon University, December 2008.

A. Krause, A. Singh, and C. Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *The Journal of Machine Learning Research*, 9:235–284, 2008.

C. Kreucher. *An Information-based Approach to Sensor Resource Allocation*. PhD thesis, University of Michigan, 2005.

V. Krishnamurthy and R. Evans. Hidden Markov Model Multiarm Bandits: A Methodology for Beam Scheduling in Multitarget Tracking. *IEEE Trans. on Signal Processing*, 49(12):2893–2908, 2001.

T. Kröger. *On-Line Trajectory Generation in Robotic Systems: Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events*. PhD thesis, Technische Universität Carolo-Wilhelmina zu Braunschweig, 2009.

E. Krotkov and R. Bajcsy. Active Vision for Reliable Ranging: Cooperating Focus, Stereo, and Vergence. *International Journal of Computer Vision (IJCV)*, 11(2), 1993.

V. Kumar, D. Rus, and S. Singh. Robot and Sensor Networks for First Responders. *IEEE Pervasive Computing*, 3(4):24–33, 2004.

R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g^2o: A General Framework for Graph Optimization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

H. Kurniawati, D. Hsu, and W. Lee. SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces. *Robotics: Science and Systems*, 2008.

H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2 edition, 2003.

C. Lampert. An Efficient Divide-and-Conquer Cascade for Nonlinear Object Detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1022–1029, 2010.

C. Lampert, M. Blaschko, and T. Hofmann. Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

X. Lan and M. Schwager. Planning Periodic Persistent Monitoring Trajectories for Sensing Robots in Gaussian Random Fields. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.

C. Laporte and T. Arbel. Efficient Discriminant Viewpoint Selection for Active Bayesian Recognition. *International Journal of Computer Vision*, 68(3), 2006.

M. Lauri, N. Atanasov, G. Pappas, and R. Ritala. Active Object Recognition via Monte Carlo Tree Search. In *IEEE Int. Conf. on Robotics and Automation (ICRA): Workshop on Beyond Geometric Constraints*, 2015.

M. Lauri and R. Ritala. Stochastic Control for Maximizing Mutual Information in Active Sensing. In *IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Robots in Homes and Industry*, 2014.

S. LaValle. Sensing and Filtering: A Fresh Perspective Based on Preimages and Information Spaces. *Foundations and Trends in Robotics*, 1(4):253–372, 2012.

W. Law. *Approximately Counting Perfect and General Matchings in Bipartite and General Graphs*. dissertation, Duke University, 2009.

J. Le Ny and G. Pappas. Sensor-based Robot Deployment Algorithms. In *IEEE Conf. on Decision and Control (CDC)*, pages 5486–5492, 2010.

J. Le Ny. *Performance Optimization for Unmanned Vehicle Systems*. PhD thesis, Massachusetts Institute of Technology, 2008.

J. Le Ny and G. Pappas. On Trajectory Optimization for Active Sensing in Gaussian Process Models. In *IEEE Conf. on Decision and Control (CDC)*, pages 6286–6292, 2009.

J. Le Ny, E. Feron, and M. Dahleh. Scheduling continuous-time kalman filters. *IEEE Trans. on Automatic Control (TAC)*, 56(6):1381–1394, 2011.

C. Lee, D. Clark, and J. Salvi. SLAM With Dynamic Targets via Single-Cluster PHD Filtering. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):543–552, 2013.

A. Lehmann, P. Gehler, and L. Van Gool. Branch&Rank: Non-Linear Object Detection. In *British Machine Vision Conference (BMVC)*, pages 8.1–8.11, 2011a.

A. Lehmann, B. Leibe, and L. Van Gool. Fast PRISM: Branch and Bound Hough Transform for Object Class Detection. *International Journal of Computer Vision (IJCV)*, 94(2):175–197, 2011b.

C. Leung, S. Huang, and G. Dissanayake. Active SLAM using Model Predictive Control and Attractor based Exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 5026–5031, 2006.

S. Li and Y. Guo. Distributed Source Seeking by Cooperative Robots: All-to-all and Limited Communications. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1107–1112, 2012.

M. Liggins, D. Hall, and J. Llinas. *Handbook of Multisensor Data Fusion*. Taylor & Francis, 2008.

M. Likhachev, G. Gordon, and S. Thrun. ARA*: Anytime A* with Provable Bounds on Sub-Optimality. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun. Anytime Dynamic A*: An Anytime, Replanning Algorithm. In *Int. Conf. on Automated Planning and Scheduling (ICAPS)*, 2005.

S. Liu and M. Krstić. Stochastic source seeking for nonholonomic unicycle. *Automatica*, 46(9): 1443–1453, 2010.

L. Ljung. Analysis of Recursive Stochastic Algorithms. *IEEE Trans. on Automatic Control*, 22(4): 551–575, 1977.

W.-K. Ma, B.-N. Vo, S. Singh, and A. Baddeley. Tracking an Unknown Time-Varying Number of Speakers Using TDOA Measurements: A Random Finite Set Approach. *IEEE Trans. on Signal Processing*, 54(9):3291–3304, 2006.

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

R. Marchant and F. Ramos. Bayesian Optimisation for Informative Continuous Path Planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

G. Mariottini and S. Roumeliotis. Active Vision-based Robot Localization and Navigation in a Visual Memory. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

S. Martínez and F. Bullo. Optimal Sensor Placement and Motion Coordination for Target Tracking. *Automatica*, 42(4):661–668, 2006.

MATLAB. *Robust Control Toolbox.* The MathWorks Inc., Natick, Massachusetts, United States, 2012.

L. Meier, J. Peschon, and R. Dressler. Optimal Control of Measurement Subsystems. *IEEE Trans. on Automatic Control*, 12(5):528–536, 1967.

F. Meyer, H. Wymeersch, M. Fröhle, and F. Hlawatsch. Distributed Estimation with Information-Seeking Control in Agent Networks. *arXiv preprint:1408.3732*, 2014.

M. Montemerlo and S. Thrun. Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 2, pages 1985–1991, 2003.

D. Moore, J. Leonard, D. Rus, and S. Teller. Robust Distributed Network Localization with Noisy Range Measurements. In *Int. Conf. on Embedded Networked Sensor Systems*, 2004.

J. Moore and B. Anderson. Coping with singular transition matrices in estimation and control stability theory. *International Journal of Control*, 31(3):571–586, 1980.

M. Morari and J. Lee. Model Predictive Control: Past, Present and Future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

M. Morelande. Joint Data Association Using Importance Sampling. In *Int. Conf. on Information Fusion*, pages 292–299, 2009.

J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo. *Random Finite Sets for Robot Mapping & SLAM*. Springer Tracts in Advanced Robotics. Springer, 2011.

M. Naghshvar and T. Javidi. Sequentiality and Adaptivity Gains in Active Hypothesis Testing. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):768–782, 2013a.

M. Naghshvar, T. Javidi, and K. Chaudhuri. Noisy Bayesian Active Learning. In *Allerton Conf. on Communication, Control, and Computing (Allerton)*, pages 1626–1633, 2012.

M. Naghshvar and T. Javidi. Active Sequential Hypothesis Testing. *The Annals of Statistics*, 41(6): 2703–2738, 2013b.

M. Naghshvar, T. Javidi, and K. Chaudhuri. Noisy Bayesian Active Learning. *arXiv:1312.2315*, 2013.

R. Newcombe. *Dense Visual SLAM.* PhD thesis, Imperial College, December 2012.

R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time Dense Surface Mapping and Tracking. In *IEEE Int. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.

J. Nguyen, N. Lawrance, and S. Sukkarieh. Nonmyopic Planning for Long-term Information Gathering with an Aerial Glider. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014.

A. Nijenhuis and H. Wilf. *Combinatorial Algorithms*. Academic Press, 1978.

D. Nistér and H. Stewénius. Scalable Recognition with a Vocabulary Tree. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168, 2006.

S. Nitinawarat and V. Veeravalli. Controlled Sensing for Sequential Multihypothesis Testing with Non-uniform Sensing Cost. In *Asilomar Conf. on Signals, Systems and Computers*, pages 1095–1099, 2013a.

S. Nitinawarat, G. Atia, and V. Veeravalli. Controlled Sensing for Multihypothesis Testing. *IEEE Trans on Automatic Control (TAC)*, 58(10):2451–2464, 2013.

S. Nitinawarat and V. Veeravalli. Controlled Sensing for Sequential Multihypothesis Testing with Controlled Markovian Observations and Non-Uniform Control Cost. *arXiv:1310.1844*, 2013b.

A. Nüchter and J. Hertzberg. Towards Semantic Maps for Mobile Robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008.

P. Ögren, E. Fiorelli, and N. Leonard. Cooperative Control of Mobile Sensor Networks. *IEEE Trans. on Automatic Control*, 49(8), 2004.

S. Oh, S. Russell, and S. Sastry. Markov Chain Monte Carlo Data Association for Multi-Target Tracking. *IEEE Trans. on Automatic Control*, 54(3):481–497, 2009.

R. Olfati-Saber. Distributed Kalman Filtering for Sensor Networks. In *Conf. on Decision and Control (CDC)*, 2007.

R. Olfati-Saber. Kalman-Consensus Filter: Optimality, Stability, and Performance. In *Joint Conf. on Decision and Control (CDC) and Chinese Control Conference*, 2009.

L. Paletta and A. Pinz. Active Object Recognition by View Integration and Reinforcement Learning. *Robotics and Autonomous Systems*, 31(1-2), 2000.

H. Pasula, S. Russell, M. Ostland, and Y. Ritov. Tracking Many Objects with Many Sensors. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, volume 2, pages 1160–1167, 1999.

M. Pedersoli, A. Vedaldi, and J. Gonzalez. A Coarse-to-Fine Approach for Fast Deformable Object Detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1353–1360, 2011.

V.-C. Pham and J.-C. Juang. A Multi-Robot, Cooperative, and Active SLAM Algorithm for Exploration. *International Journal of Innovative Computing, Information and Control*, 9(6), 2013.

R. Pito. A Solution to the Next Best View Problem for Automated Surface Acquisition. *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, 21(10), 1999.

C. Potthast and G. Sukhatme. A Probabilistic Framework for Next Best View Estimation in a Cluttered Environment. *Journal of Visual Communication and Image Representation*, 25(1), 2014.

S. Prentice and N. Roy. The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance. *The International Journal of Robotics Research*, 28(11-12):1448–1465, 2009.

J. Principe. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Springer, 2010.

N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-Free Distributed Localization in Sensor Networks. In *Int. Conf. on Embedded Networked Sensor Systems*, pages 340–341, 2003.

A. Pronobis. *Semantic Mapping with Mobile Robots*. PhD thesis, KTH Royal Institute of Technology, 2011.

J. Quiñonero-Candela and C. Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6, 2005.

E. Rafajlowicz. Optimum Choice of Moving Sensor Trajectories for Distributed-Parameter System Identification. *International Journal of Control*, 43(5):1441–1451, 1986.

K. Rahnama Rad and A. Tahbaz-Salehi. Distributed Parameter Estimation in Networks. In *Conf. on Decision and Control (CDC)*, 2010.

E. Rahtu, J. Kannala, and M. Blaschko. Learning a Category Independent Object Detection Cascade. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1052–1059, 2011.

J. Rawlings and D. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

A. Rényi. On measures of entropy and information. In *Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561, 1961.

W. Rugh. *Linear System Theory*. Prentice-Hall, 2 edition, 1996.

R. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Technische Universität München, 2009.

A. Ryan and J. Karl Hedrick. Particle Filter Based Information-theoretic Active Sensing. *Robotics and Autonomous Systems*, 58(5):574–584, 2010.

P. Rybski, S. Stoeter, M. Erickson, M. Gini, D. Hougen, and N. Papanikolopoulos. A Team of Robotic Agents for Surveillance. In *Int. Conf. on Autonomous Agents*, pages 9–16, 2000.

H. Ryser. *Combinatorial Mathematics*. Carus Mathematical Monographs # 14. Mathematical Association of America, 1963.

B. Sankaran. Sequential Hypothesis Testing for Next Best View Estimation. Master's thesis, University of Pennsylvania, 2012.

B. Sapp, A. Toshev, and B. Taskar. Cascaded Models for Articulated Pose Estimation. In *European Conference on Computer Vision (ECCV)*, volume 6312, pages 406–420. Springer, 2010.

M. Schwager, P. Dames, D. Rus, and V. Kumar. A Multi-Robot Control Policy for Information Gathering in the Presence of Unknown Hazards. In *International Symposium on Robotics Research*, pages 17–33, 2011.

S. Se, D. Lowe, and J. Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.

S. Shahrampour, A. Rakhlin, and A. Jadbabaie. Online Learning of Dynamic Parameters in Social Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

H. Sidenbladh and S.-L. Wirkander. Tracking Random Sets of Vehicles in Terrain. In *Computer Vision and Pattern Recognition Workshop*, volume 9, pages 98–98, June 2003.

D. Silver and J. Veness. Monte-Carlo Planning in Large POMDPs. In *Neural Information Processing Systems (NIPS) 23*, pages 2164–2172, 2010.

R. Sim and N. Roy. Global A-Optimal Robot Exploration in SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 661–666, 2005.

A. Singh, A. Krause, C. Guestrin, and W. Kaiser. Efficient Informative Sensing Using Multiple Robots. *Journal of Artificial Intelligence Research (JAIR)*, 34(1):707–755, 2009a.

A. Singh, A. Krause, and W. Kaiser. Nonmyopic Adaptive Informative Path Planning for Multiple Robots. In *21st Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 1843–1850, 2009b.

A. So and Y. Ye. Theory of semidefinite programming for Sensor Network Localization. *Mathematical Programming*, 109(2-3):367–384, 2007.

E. Sommerlade and I. Reid. Information Theoretic Active Scene Exploration. In *IEEE Computer Vision and Pattern Recognition*, 2008.

J. Spall. *Introduction to Stochastic Search and Optimization*. Wiley, 2003.

D. Spanos, R. Olfati-Saber, and R. Murray. Dynamic Consensus on Mobile Networks. *16th IFAC World Congress*, 2005.

M. Sridharan, J. Wyatt, and R. Dearden. Planning to See: A Hierarchical Approach to Planning Visual Actions on a Robot using POMDPs. *Artificial Intelligence*, 174(11), 2010.

M. Stanković and D. Stipanović. Extremum Seeking under Stochastic Noise and Applications to Mobile Sensors. *Automatica*, 46(8), 2010.

E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric Belief Propagation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 605–612, 2003.

N. Sünderhauf and P. Protzel. BRIEF-Gist - Closing the Loop by Simple Means. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1234–1241, 2011.

R. Sznitman, C. Becker, F. Fleuret, and P. Fua. Fast Object Detection with Entropy-Driven Evaluation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3270–3277, 2013.

J. Theiler and J. Alper. On the choice of random directions for stochastic approximation algorithms. *IEEE Trans. on Automatic Control*, 51(3):476–481, 2006.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press Cambridge, 2005.

P. Tokekar, E. Branson, J. Vander Hook, and V. Isler. Tracking Aquatic Invaders: Autonomous Robots for Monitoring Invasive Fish. *IEEE Robotics and Automation Magazine*, 20(3):33–41, 2013.

P. Tokekar. *Placement and Motion Planning Algorithms for Robotic Sensing Systems*. PhD thesis, University of Minnesota, October 2014.

B. Tovar, L. Muñoz-Gómez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy, and S. Hutchinson. Planning Exploration Strategies for Simultaneous Localization and Mapping. *Robotics and Autonomous Systems*, 54(4):314–331, 2006.

N. Trawny and S. I. Roumeliotis. Indirect Kalman Filter for 3D Attitude Estimation. Technical Report 2005-002, University of Minnesota, Dept. of Comp. Sci. and Eng., March 2005.

R. Tron and R. Vidal. Distributed Image-Based 3-D Localization in Camera Sensor Networks. In *Joint Conf. on Decision and Control (CDC) and Chinese Control Conference*, 2009.

L. Valente, Y.-H. Tsai, and S. Soatto. Information-Seeking Control Under Visibility-Based Uncertainty. *Journal of Mathematical Imaging and Vision*, 48(2):339–358, 2014.

L. Valiant. The Complexity of Computing the Permanent. *Theoretical Computer Science*, 8(2): 189–1201, 1979.

J. Vander Hook, P. Tokekar, and V. Isler. Cautious Greedy Strategy for Bearing-only Active Localization: Analysis and Field Experiments. *Journal of Field Robotics*, 31(2):296–318, 2014.

J. Velez, G. Hemann, A. Huang, I. Posner, and N. Roy. Modelling Observation Correlations for Active Exploration and Robust Object Detection. *Journal of Artificial Intelligence Research*, 44, 2012.

T. Vidal-Calleja, A. Davison, J. Andrade-Cetto, and D. Murray. Active Control for Single Camera SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1930–1936, 2006.

P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

M. Vitus, W. Zhang, A. Abate, J. Hu, and C. Tomlin. On Efficient Sensor Scheduling for Linear Dynamical Systems. *Automatica*, 48(10), 2012.

J. Wang, H. Zha, and R. Cipolla. Coarse-to-Fine Vision-Based Localization by Indexing Scale-Invariant Features. *IEEE Trans. on Systems, Man, and Cybernetics*, 36(2):413–422, 2006.

D. Weiss, B. Sapp, and B. Taskar. Structured Prediction Cascades. *arXiv:1208.3279*, 2012.

J. Williams. *Information Theoretic Sensor Management.* PhD thesis, Massachusetts Institute of Technology, February 2007.

J. Wolf, W. Burgard, and H. Burkhardt. Robust Vision-based Localization by Combining an Image Retrieval System with Monte Carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.

L. Wong, L. Kaelbling, and T. Lozano-Pérez. Data Assocation for Semantic World Modeling from Partial Views. In *International Symposium on Robotics Research (ISRR)*, 2013.

T. Wu and S.-C. Zhu. Learning Near-Optimal Cost-Sensitive Decision Policy for Object Detection. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 753–760, 2013.

W. Wu and F. Zhang. Experimental Validation of Source Seeking with a Switching Strategy. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3835–3840, 2011.

B. Yamauchi. A Frontier-Based Approach for Autonomous Exploration. In *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 146–151, 1997.

C. Yi, I. H. Suh, G. H. Lim, and B.-U. Choi. Active-Semantic Localization with a Single Consumer-Grade Camera. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 2161–2166, 2009.

C. Zhang, D. Arnold, N. Ghods, A. Siranosian, and M. Krstić. Source seeking with non-holonomic unicycle without position measurement and with tuning of forward velocity. *Sys. and Control Letters*, 56(3):245–252, 2007.

F. Zhang, H. Stahle, A. Gaschler, C. Buckl, and A. Knoll. Single Camera Visual Odometry Based on Random Finite Set Statistics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 559–566, 2012.

F. Zhang and N. Leonard. Cooperative Filters and Control for Cooperative Exploration. *IEEE Trans. on Automatic Control*, 55(3):650–663, 2010.

Z. Zhang, J. Warrell, and P. Torr. Proposal Generation for Object Detection using Cascaded Ranking SVMs. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1497–1504, 2011.

L. Zhao, W. Zhang, J. Hu, A. Abate, and C. Tomlin. On the Optimal Solutions of the Infinite-Horizon Linear Sensor Scheduling Problem. *IEEE Trans. on Automatic Control*, 59(10):2825–2830, 2014.

K. Zhou. *Active Sensing with Applications to Mobile Robotics*. PhD thesis, University of Minnesota, June 2012.

K. Zhou and S. Roumeliotis. Multirobot Active Target Tracking With Combinations of Relative Observations. *IEEE Trans. on Robotics (TRO)*, 27(4):678–695, 2011.

M. Zhu, N. Atanasov, G. Pappas, and K. Daniilidis. Active Deformable Part Models Inference. In *European Conf. on Computer Vision (ECCV)*, volume 8695, pages 281–296, 2014.