2016

GPS-denied multi-agent localization and terrain classification for autonomous parafoil systems

BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

GPS-DENIED MULTI-AGENT LOCALIZATION AND TERRAIN CLASSIFICATION FOR AUTONOMOUS PARAFOIL SYSTEMS

by

EVE LAW

B.S., Harvard University, 2011
M.S., Boston University, 2013

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2016

Approved by

First Reader

John Baillieul, PhD
Distinguished Professor of Mechanical Engineering
Distinguished Professor of Electrical Engineering
Distinguished Professor of Systems Engineering

Second Reader

Sean B. Andersson, PhD
Associate Professor of Mechanical Engineering
Associate Professor of Systems Engineering

Third Reader

Roberto Tron, PhD
Assistant Professor of Mechanical Engineering

Fourth Reader

Sang (Peter) Chin, PhD
Research Professor of Computer Science

Acknowledgments

I would like to thank:

- The Charles Stark Draper Laboratory for funding this research under the Guided Airdrop project, sponsored by Natick Soldier Research Development and Engineering Center
- The entire Airdrop team, past and present, for their guidance and support in this endeavor, with special thanks to Josh Torgerson, Chris Dever, Rob Truax, and Matthew Neave
- Professor John Baillieul for his guidance and expertise
- Professors Sean Andersson, Roberto Tron, and Peter Chin for serving on my thesis committee, as well as Mac Schwager for his inspiration in pursuing this topic
- My family for the great privilege of their love and support

GPS-DENIED MULTI-AGENT LOCALIZATION AND TERRAIN CLASSIFICATION FOR AUTONOMOUS PARAFOIL SYSTEMS

EVE LAW

Boston University, College of Engineering, 2016

Major Professor: John Baillieul, PhD
Distinguished Professor of Mechanical Engineering
Distinguished Professor of Electrical Engineering
Distinguished Professor of Systems Engineering

ABSTRACT

Guided airdrop parafoil systems depend on GPS for localization and landing. In some scenarios, GPS may be unreliable (jammed, spoofed, or disabled), or unavailable (indoor, or extraterrestrial environments). In the context of guided parafoils, landing locations for each system must be pre-programmed manually with global coordinates, which may be inaccurate or outdated, and offer no in-flight adaptability. Parafoil systems in particular have constrained motion, communication, and on-board computation and storage capabilities, and must operate in harsh conditions. These constraints necessitate a comprehensive approach to address the fundamental limitations of these systems when GPS cannot be used reliably. A novel and minimalist approach to visual navigation and multi-agent communication using semantic machine learning classification and geometric constraints is introduced. This approach enables localization and landing site identification for multiple communicating parafoil systems deployed in GPS-denied environments.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Guided airdrop systems present numerous navigation and guidance challenges, including finite flight-time, a need to land in payload-survivable locations, and time-delayed nonholonomic dynamics. These systems currently rely primarily on GPS for navigation, guidance, and landing. Each system must be equipped with military-spec GPS which can fail, be jammed or spoofed, or may not be available in future scenarios. Landing locations for each system must be pre-programmed manually with global coordinates, which may be inaccurate or outdated, and offer no in-flight adaptability. By specifying desired landing sites in global coordinates, accurate landing thus requires global localization.

The objective of this work is to address these limitations and constraints with navigation algorithms enabling localization and landing site identification for multiple communicating parafoil systems with flight-time and communication constraints deployed in GPS-denied environments.

The recent emergence of powerful vision sensors and processing capabilities presents numerous possibilities for overcoming some of the limitations imposed by a GPS-denied scenario. This work seeks to further the state of the art in visual localization by harnessing recently developed scene recognition and understanding machine learning algorithms to rethink how vision can be used for navigation.

Rather than focus on vision and multiple agents as complicated substitutes for GPS and more powerful single agents, this work focuses on demonstrating the true potential of each:

better, not more convoluted.

Machine learning based visual understanding can enable a novel method of localization, and naturally combines with multi-agent coordination to improve landing optimally, with or without global localization as well. This adds significant capabilities to current parafoil systems, enabling a host of new and expanded deployment scenarios.

The significance of this research is a reconsideration of navigation paradigms for parafoil systems, and autonomous vehicles generally, especially when operating in GPS-denied environments. This has applications in situations ranging from indoor localization to space exploration, enabling new modalities and capabilities for existing and future autonomous systems.

Sections 1.1-1.3 outline some preliminary assumptions and scenarios, as well the contributions of this work. Chapter 2 provides background with a brief literature review, highlighting remaining challenges. The general approach pursued is proposed and introduced by example in Chapter 3. Mathematical foundations and intuitions are presented in Chapter 4. Chapters 5 and 6 apply the proposed approaches to in-flight imagery. Chapter 7 concludes with discussion of potential future work.

1.1 Environment Sensing and Communication Assumptions

In a GPS-denied setting, other sensors must provide information about the environment, sometimes termed extraperceptive sensing. Similarly, for any meaningful multi-agent approach, some form of communication and sensing protocol between agents must exist. For this work, vision assumes the primary role of extraperceptive sensing, and some type of radio communication and ranging are assumed available for coordination between agents. Neither of these assumptions should be seen as limiting, in the sense that both represent well-established capabilities of unmanned autonomous systems, and can be substituted with

other specifics without loss of generality (i.e. terrain elevation sensing, laser scanning, radar, line-of-sight communication).

1.2 Scenarios

To fix ideas, specific scenarios which current parafoil navigation algorithms cannot address are outlined. These include both single and multi-agent cases, working with a potentially very large or outdated map prior of their environment, or perhaps no map at all. The objective is to land in the best location possible (a landing site cost function will be defined formally in Section 3.5). The proposed work seeks to address cases such as:

- Multiple-agents, map prior
  - Global localization possible, though not required
  - Communication with neighbors to localize, identify globally reachable landing site for group based on observations and map matching
- Single-agent, no map prior
  - Cannot globally localize
  - Searches for suitable landing sites
- Mutliple-agents, no map prior
  - Cannot globally localize
  - Communication with neighbors to collectively identify best landing site

Each of these scenarios represent overlapping challenges that state of the art parafoil navigation algorithms are not equipped to handle.

1.3 Contributions

The primary contribution of this work is the introduction of a novel approach to visual localization and landing in a GPS-denied setting. This new approach combines machine

learning-based semantic classification together with minimalist multi-agent communication to address navigation in profoundly perceptually aliased environments. Harnessing semantic classification to represent imagery in a compressed manner, combined with scalar geometric distance constraints from other agents, produces a minimalist, statistically consistent, and robust approach to multi-agent localization and terrain classification that can account for non-overlapping viewpoints. This work represents a clear demonstration of how multi-agent visual localization can be more capable, rather than merely more complex, than reliance on GPS alone. The algorithms developed add new capabilities both to the specific application of guided parafoil navigation, as well as GPS-denied scenarios more broadly. Background in the areas of visual localization and multi-agent coordination is helpful for understanding the limitations of the current state of the art, which is the focus of Chapter 2.

Chapter 2

Background

The problems of GPS-denied localization, along with landing-site identification, are far from new. The following sections describe some of the leading approaches to these problems, pointing out some of the strengths and limitations as applied in a multi-agent guided parafoil context.

2.1 GPS-denied Collective Localization: State of the Art

Single and multi-agent robot localization, typically referred to as the kidnapped or lost robot, or simultaneous localization and mapping (SLAM) problem, has been a subject of study for at least the past few decades. What follows is a brief overview with emphasis on more recent work relating to multi-agent contexts. Recent reviews of multi-agent work are [9] and [61].

Solving the SLAM problem can be roughly broken down into the following steps:

1. Odometry: tracking of robot body-frame pose over time
2. Data association: matching of sensor data (e.g. vision, laser) from environment to previous measurements (loop closures) or known values (landmarks)
3. Estimation of robot pose and map posterior: nonlinear optimization of odometry, loop closure, and landmark constraints

The first two steps are typically referred to as front-end, i.e. data collection, while the last esimation step is the back-end. In this work, the latter two stages will be emphasized.

2.2 Odometry

Odometry estimation has made great strides in the last decade, with the work in [38] and follow-up papers (e.g. [35]) demonstrating excellent performance in maintaining small tracking drift over time using inertial and visual sensing in the presence of GPS blackouts. The development of an efficient and effective manner for updating previous measurements with clever use of vector null-spaces combined into an approach termed MSCKF (Multi-State Constraint Kalman Filter) enables real-time, nearly drift-free visual odometry based on feature tracking that combines inputs from inertial measurement units (IMU) as well as global pose information (whether from GPS or otherwise) seamlessly. While challenges remain in this area–specifically when encountering repeating-pattern visual areas, or remaining motionless–existing methods, whether MSCKF or more conventional sliding window filters, are very capable [15]. Availability of excellent odometry is thus a reasonable assumption, and will be relied upon in Chapters 5 and 6.

2.3 Data Association

The state of the art in single-agent data association is Fast Appearance-Based Mapping (FAB-MAP 2.0) [18] which combines a Bayesian approach to a discretization of space together with a bag of visual words approach. A visual bag of words approach generates features to classify a data-set of imagery, based on the natural language processing bag of words technique, which captures the multiplicity of data elements (words) in a multiset (bag). Very impressive results are presented showing accurate place recognition along streets and highways. In [8], GPS-denied localization in an urban environment is achieved by a ground vehicle using visual and inertial sensing with an on-board street map represented as a directed graph, estimating location based on the observed sequence of traversing streets using a Gaussian mixture model. A key idea in both of these works is assuming a discretized and structured map environment. What remains unclear is how well these method scale to massive maps, different camera viewpoints, and branching data (i.e. free space, rather than street routes).

Ensuring multiple agents associate data correctly when overlapping landmarks are observed is another aspect of the problem, addressed in e.g. [21] [37] using triangulation and consensus, respectively. Both these approaches require sufficient and regular observation overlap to disambiguate landmarks reliably, however. A successful approach for combining estimates of multiple agents with non-overlapping landmark observations in a GPS-denied setting has not been reported to date.

More generally, the problem of visual or perceptual aliasing, where multiple images appear similar but are in fact different, leading to spurious loop closure/landmark identifications [17], remains incompletely solved.

2.4 Estimation of Pose and Map

The estimation of pose and map (back-end) given measurements is typically considered in a Bayesian framework [2], and three noteworthy approaches are 1) extended Kalman filters (EKFs), 2) factor graphs, and 3) particle filters. In each framework, the issue of maintaining consistency when multiple agents communicate arises. Consistency, a term precisely defined in statistics regarding the convergence of an estimator to the true distribution, is used in a multi-agent localization context as a concern for estimators failing to account for dependence of estimates among communicating agents.

Work representative of these three prevailing approaches is reviewed, with advantages and drawbacks outlined.

2.4.1 Extended Kalman Filter (EKF)

State of the approaches to odometry rely on EKFs, which are thus natural frameworks for solving the larger SLAM problem as well. Briefly, EKFs are Kalman Filters applied to nonlinear systems through a relinearization around current state estimates in real-time as a filter is running. The work of Roumeliotis over the past decades sets the benchmark for

EKF approaches. In [46], localization in a kidnapped robot scenario considers the issues of non-unique landmarks. Vision-aided planetary landing is considered in [58] and [39], where an approach for matching mapped landmarks suitable for an EKF is detailed. Incorporation of terrain elevation data with vision has been considered in [54], based on terrain contour matching (TERCOM), described in [55], where known elevation data is matched to on-board altitude measurements (commonly used in terminal missile guidance).

In a multi-agent setting, the term collective localization (CL) is introduced in [47], where consistency and independence issues are described, and a more complete treatment is given in [48]. Distributed approaches using maximum a posteriori (MAP) estimation [41] and inherently conservative covariance intersection in multiple flavors, e.g. [10] [4], have also been presented as methods for ensuring consistency.

The main drawbacks to EKF approaches concern the imposition of Gaussian structure on the problem, the computationally expensive and sometimes challenging relinearization step, and a generally cumbersome form when tasked with the larger SLAM problem. Particular examples that have been optimized for specific applications work well, and it remains the de facto choice in practical estimation. However, the specter of divergence and inconsistency without good initialization makes it appear less than ideal for the current work.

2.4.2 Factor Graph

In recent years, an approach has emerged that enables potentially better and faster real-time solutions to SLAM by exploiting the sparsity of a factor graph [31] formulation of the problem (Figure 2·1). Some of the more recent influential work in this direction is [29] and [30] detailing incremental approaches to using a factor graph back-end to solve for computationally and dimensionally challenging loop closure and landmark matchings, and providing updates to a concurrently running EKF considering only odometry measurements. In a multi-agent context, a factor graph framework has been used in distributed data fusion

by having all agents transmit their own factor graphs [20] [21] [19], or some kind of condensed version [34]. Consistency issues are addressed using anti-factors, which requires extensive book-keeping of all previous communications. Factor graph robustness algorithms provide an approach to addressing some instances of perceptual aliasing as well, e.g. [24], [43].

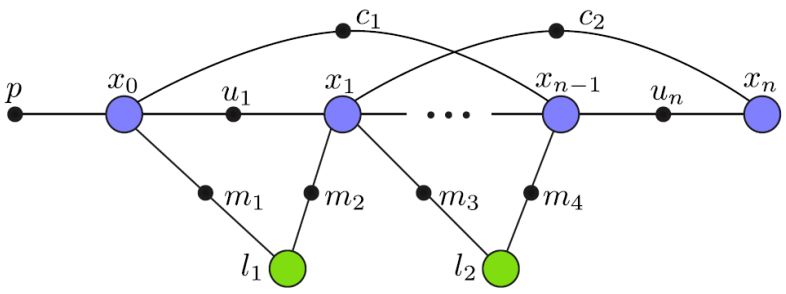


Figure 2·1: SLAM considered as a factor graph, where larger circles are variable nodes, and measurement factors are small circles. Odometry factors u, loop closures c, and landmark constraints m are shown (From [29]). By combining these constraints into a graph representation the inherent sparsity of the problem emerges, enabling more efficient calculation.

While factor graphs represent state of the art approaches to solving SLAM, the case for use in a multi-agent scenario, while a topic of research, requires broad assumptions about the level and bandwidth of available communication. Recent work has attempted to consider these situations with more realistic constraints (e.g. [14]), but the great advantages of representing information in a graph are diminished when multiple agents must somehow share and incorporate potentially different graphs together.

2.4.3 Particle Filter

A particle filter approach using Bayesian updates and Monte Carlo sampling has been pursued for both single [22] and multiple agents [23], where consistency is maintained with heuristics. This method was in vogue at the turn of the century, but has since fallen out of favor with the emergence of efficient and effective EKFs, as well as the relatively limited opportunity for closed-form mathematical results. The flexibility of being able to represent

any distribution, rather than assuming Gaussian structure, remains promising, and appears to be worth reconsideration.

In summary, there is no shortage of work in estimation for the kidnapped robot or SLAM problems generally, and multi-agent estimation in particular. As of yet, however, there have not been clear examples of algorithms with realistic constraints demonstrating how multiple agents can be better, or more capable, than a single agent working alone.

2.5 Active SLAM, or Exploration vs Exploitation

Another topic of active research is choosing guidance strategies to optimize localization and/or mapping, also known as the active SLAM problem. Promising strategies rely on information theoretic approaches, including following entropy or mutual information gradients (e.g. [7] [50] [5] [28]). These approaches typically assume a discretized occupancy grid (a binary variable representing whether a location is occupied by an obstacle or free space) and scalar field type environment. While often computationally expensive, they represent a principled optimal search strategy in suitably structured environments.

2.6 Parafoil Guidance and Landing Site Identification

Typical autonomous parafoil guidance is composed of four main stages: (1) homing to the ground coordinates of the target, (2) reducing altitude or energy management, (3) path planning for terminal guidance, and (4) flaring to soften landing. Strategies of this kind, described in [11] [44] [59] assume the parafoil is programmed with the target landing site at the beginning of flight, and can navigate to the airspace above the ground coordinates of the target and loiter there while altitude is reduced. Recent research has looked at rethinking this paradigm for the purposes of dealing with high wind scenarios [12], but relaxing the assumption of knowing the final landing site from the outset has not been investigated to date. With respect to multi-agent parafoil guidance, work has focused on reactive collision avoidance and wind estimate sharing [45], but no comprehensive approach to overall local-

ization and landing site identification or guidance has been reported.

Previous landing site identification work concerning detection for powered landing craft with loitering capabilities relies on pre-constructed concrete pads, and finding distinct characteristics of brightness, area, and solidity [56]. While the principle of detecting landing sites using these parameters is promising, specifying and detecting landing sites more broadly, particularly in a parafoil context with time constraints, remains an open challenge.

2.7 Remaining Challenges

To summarize, in achieving the proposed project objectives, state of the art approaches do not appear to adequately address the following issues:

- Scaling to large, branching map data, matching different viewpoints
- Challenging visual/perceptual aliasing environments
- Exploiting additional capabilities vision provides over GPS
- Multi-agent sharing of estimates and map representations without overlap (multi-agent data association)
- Realistic communication protocols/constraints
- Demonstrable advantages of multi-agent estimation
- Landing site identification under time constraints considered broadly

Chapter 3 details the proposed approach of this work to address some of these open problems.

Chapter 3

Proposed Approaches

In addressing the challenges outlined in Section 2.7, the following broad ideas are presented and described. Each will be developed further in succeeding chapters.

- Visual localization should be better, not just more complicated:
  - Classify environment using scene recognition/understanding (Section 3.2, 3.3)
- Solve localization and landing site decision problem simultaneously (Section 3.4)
- Multi-agent scenario should be better, not just more complicated (Section 3.6)
  - Use scalar distance information from other agents (e.g. signal strength)
  - Minimalist assumptions on information availability from other agents

3.1 Assumptions

One of the major issues encountered in reading both localization and multi-agent work is the hidden assumptions. Common unstated, but substantial, assumptions include having already established a common reference frame (such as knowing global heading), extensive communication capabilities where each agent knows each other agent's complete state every time step, or some kind of central processing location.

The approach of this work is to begin with very minimal assumptions about each agent's capabilities and to state them explicitly. In particular, each agent is assumed to have the following:

- Altitude estimation. Whether based on barometer, radar/sonar to the ground, or initial knowledge of altitude and estimation based on elapsed time and glide slope knowledge.

- Nadir pose estimation. Knowledge of the ground plane normal, to be capable of projecting imagery directly downward. Consistent with inertial measurement unit (IMU) measurements.

- Basic radio communication and radar ranging capabilities, to estimate the scalar distance which measurements from other agents originate from. Simulations will further assume communication between all agents is possible every time step, but this is used to demonstrate the capabilities in that scenario, rather than being required. Indeed, an essential aspect of the algorithms presented will be robustness, by design, to intermittent or failing communications, and relatively low bandwidth requirements.

In essence, this means that of the 6 degrees of freedom, global altitude, pitch, and roll angles are known to within estimation errors. The remaining 3 degrees of freedom, namely global 2-D location (latitude and longitude) and heading (with respect to North, for instance) in the plane are unknown. Body-frame translations and yaw are assumed to be known to within estimation error (as would be available with IMU), and can be used for relative odometry.

The purpose of these assumptions is to enable an agent to properly project and scale camera imagery to the ground plane, and have a general idea of how far away other agents are located, and should not be seen as overly limiting. These assumptions are consistent with real-world guided parafoil capabilities.

Regarding communications, the agents operate in a fully decentralized manner. There is no central processor to fuse their measurements together, and no pre-existing local network upon which they can communicate. There are also no fixed-location beacons in the envi-

ronment. In the terminology of [61], this is a decentralized, active implicit communication model.

3.2 Scene Recognition / Understanding

A major challenge with visual navigation is the unique identification of locations. In essence, many places look the same. State of the art approaches to this problem, typically referred to as the place recognition problem, rely on building high dimensional image descriptors, with the goal of capturing enough detail to uniquely identify locations. There have been impressive results reported in particular circumstances, detailed in Section 2.3. The fundamental problem of accurately distinguishing one very visually similar location from another, or recognizing the same location from different viewpoints, or observed under different lighting or weather conditions, remains a major hurdle. This problem is often referred to as perceptual aliasing: images of two (or more) different locations appear to be the same, or two (or more) images of the same location appear different [17]. There are fundamental limitations to uniquely identifying locations visually, such as the middle of an ocean or desert, as concrete examples.

Moreover, if attempting to match observed images to a geotagged map to globally localize, the entire map must be stored and analyzed using these same high dimensional descriptors. While computing power and storage continue to dramatically increase year after year, the curse of dimensionality makes an approach for storing and comparing images to large maps in this manner appear infeasible for real-time navigation for the foreseeable future. Approaches such those detailed in [18] rely on approximation tricks to structure these high dimensional classifiers and Bayesian updates in ways to make the processes more real-time capable, but do not address the fundamental underlying problems.

Most importantly, the conventional place recognition approach to navigation does not attempt to comprehend the observed scene. High dimensional image classifiers do not have

any semantic meaning, and can only be used as comparisons against other images classified in the same manner. In essence, this is navigation by seeing without understanding.

The implications of this are dramatic. In a parafoil context, landing sites must be specified in global coordinates, requiring global localization in all cases, lacking the flexibility to adapt to conditions on the ground. There is no way to semantically specify or locate landing locations if the imagery being processed is not understood.

In addition, when communicating with other agents, vast amounts of data must be passed in order to convey what each agent is seeing and to integrate that information in a consistent manner (Section 2.4). There is no universally understandable language in which agents can communicate, and systems with different sensing modalities and capabilities cannot easily share their information.

This work proposes a different approach to these challenges, predicated on a simple idea. While multiple images of one kind of terrain may be indistinguishable from others, images of different terrain types–sand and forest, for instance–can be uniquely distinguished from each other.

Rather than fight perceptual aliasing, embrace and exploit it. Rather than storing maps in computationally unwieldy forms, use a condensed and compressed approach, with implications for multi-agent coordination as well. Rather than navigate by blindly seeing without understanding, requiring global localization for all missions, instead use semantically meaningful visual recognition algorithms (Section 3.3). Combined with geometric constraints, such as odometry and measurements from other agents, particular locations can then be determined probabilistically (Sections 3.4 and 3.6).

These ideas can be seen as a method to complement existing place recognition algorithms,

serving as a fallback for circumstances where no reliable GPS is available, in addition to providing new capabilities and methods for addressing the motivating scenarios described in Section 1.2.

3.2.1 Visual Richness

To fix ideas, a new concept termed visual richness (VR) is introduced here to describe the various kinds of environments one may encounter, and help define the different problems being addressed.

Define a function for visual richness VR: $M \rightarrow [0, 1]$ where M is a map of an environment. Let VR $= 0$ be defined as a completely homogenous environment (such as an ocean or desert), where there are no visually unique landmarks, and thus visual localization is impossible. Similarly, let VR $= 1$ define an environment where every location is visually unique (for instance a labeled map of a city), and a single image is sufficient for determining location immediately without ambiguity. This notion can be thought of as either intrinsic to the environment itself, or to the capabilities of sensors to observe and measure the environment (perception is reality).

The approach of most place recognition algorithms is to build intricate databases with features of high dimension to try to push environments as close to a VR $= 1$ as possible. High dimensional whole-image features using intermediate layers of a neural network (as in [57] [49]) are examples of this approach, as is the visual bag of words approach from SURF features used in FAB-MAP [18]. In the scenario of genuinely visually unique environments, where VR approaches 1, this is a reasonable undertaking.

In the real world, particularly in natural environments, there exist many circumstances where VR is not close to 1. To give a concrete example, consider an environment of 100 discrete locations that can only be categorized into one of two unique categories A and B,

distributed evenly (50/50): for instance, an open clearing in a forest. A noise-free sample image from this environment will allow an observer to narrow down their location in this environment by half, either forest or open area (ignoring borders for the moment). Similarly, if things are divided into three categories, they would reduce their uncertainty by 2/3, etc. These would then be examples of VR $= \frac{1}{2}$ and VR $= \frac{2}{3}$, respectively.

When a second observation is made, an explicit link to the first is necessary to further reduce uncertainty. In a sense, rather than search for a single tile of A, now one has narrowed things to two contiguous tiles of type A. Similarly, with a measurement transmitted from another agent a known distance away, one has another geometric constraint by which to narrow down the possible tiles of type A one is currently located in. Noise in these observations of course complicates things, but a Bayesian framework naturally handles this by reweighting location probabilities appropriately. What should be clear, however, is that these cases represent a fundamentally different type of visual environment than ones with VR approaching 1. By very definition, no purely visual approach could ever localize here without additional information. The proposal described in this work is designed to address these scenarios where VR is substantially far from 1, where structural perceptual aliasing demands another approach.

More broadly, one could consider that some terrain types fit together. As opposed to the visual bag of words approach, which simply encodes multiplicities and ignores grammar and other interconnections of words (features), one could explicitly link semantic terrain types based on observations imposed by nature. For instance, in the middle of a desert, while there might be the occasional oasis, generally there cannot be regularly spaced tiles of water; the opposite may be true generally as well, i.e. the middle of the ocean is not usually sprinkled with small land masses. To be sure, there are occurrences of similar types of locations, whether man-made irrigation, or certain island formations. But in effect, by being rare, those could be considered closer to VR of 1, since there are only so many places

in the world of that variety that exist. Scaling plays a role as well: visually knowing with certainty that one is located in the Hawaiin Islands, for instance, is not the same as knowing precisely where one is located. Exploring the relationship between VR and Visual Saliency [26] could be pursued as well, where particular parts of an environment are more distinctive than others. Thus VR is both a space and scale (and possibly time) dependent quantity that may not, in general, be computable. For particular circumstances, it helpfully distinguishes different types of environments, and could be a topic of further study in its own right.

3.3 Scene Recognition: Convolutional Neural Networks (CNNs)

Scene recognition and understanding represents a different approach to image analysis than traditional visual classifiers. Rather than detect and classify images using specific definitions of features, such as SIFT [36] or SURF [6], consider the entire image (or objects contained therein). The concept of whole image classification in a semantic sense has been a topic of study for at least the last decade and a half, the GIST descriptor [42] being the most noteworthy method. Navigation using this descriptor has been explored in an urban environment [40]. In addition, the concept of a spatial semantic hierarchy, introduced in [32] is another approach to navigating within environments based on identifying particular types of objects, such as chairs or tables.

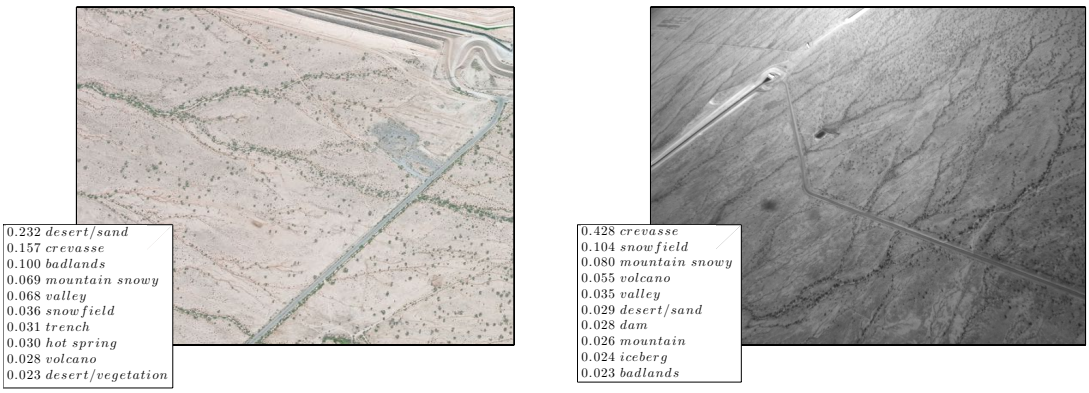
More recently, advances in machine learning (ML) have led to a new generation of image classification methods. Convolutional Neural Networks (CNNs) represent a state of the art approach in ML for object and scene recognition [27] [62]. CNNs have been successfully applied to many data analysis fields, and are particularly effective in vision applications.

Technically, a CNN represents a function taking images (pixels) as inputs, producing category probabilities as outputs, parameterized by a series of convolutional filters which are found using stochastic gradient descent to optimize a loss function on labeled training data. The resulting nets have multiple layers of different types, which can be adjusted based on

particular applications, and canonical forms have emerged as effective in recent work [27] [62]. While the idea of CNNs is not new, the ability to harness GPU processing for efficient classification is a recent development, leading to a dramatic increase in popularity.

An approach to solving the problem of identifying the location of images globally based solely on image pixels is reported in [60] by choosing the outputs of a CNN to be discretized locations in a map of Earth. Drawing inspiration from the pioneering work in im2GPS [25], a new CNN (termed PlaNET) was developed and trained based on 126 million images from all over the web. This direct approach of creating a CNN from pixels to geolocation is quite ambitious, and while the reported accuracy to street level (1 km) was 8.4%, the continent level accuracy (2500 km) of 71.3% is very impressive given the great variability in imagery used and the goal of encompassing the entire planet. Of note, the authors chose to represent the earth using discrete cells of varying and adaptable size, dependent on the number of input images from those regions. The authors also investigated using information from multiple images contained in the same album to try to improve localization, but found their database to be too convoluted to improve overall results (images grouped together do not necessarily come from the same location). This reflects a general issue with machine learning approaches: asking them to solve harder and harder problems directly, entirely in a black-box fashion, may be less useful than improving existing networks. Using well-established networks in more efficient and interesting ways, where quantifiable metrics for how they perform are available, enables seeing the true trade-offs of different approaches, rather than repeatedly introducing entirely new stand-alone networks.

CNNs have also been applied to conventional place recognition/pose estimation with promising results [57] [49], but typically rely on higher dimensional layers of the network, rather than the final, categorized layer. These approaches use layers of CNNs as another black-box, high dimensional image descriptor. Other approaches consider machine learning as a way of augmenting traditional place recognition, e.g. [1]. Visual navigation based on scene



(a) (b)

Figure 3·1: Examples of CNN classification for same location. Note the shape of the roadway intersection in both images. For more context, the satellite tile in (a) is tile 63 from the larger satellite map depicted in Figure 5.2.

recognition, rather than specific place recognition, remains an open problem.

There are numerous CNNs being developed, and one of particular interest is the Places-CNN model [62]. It is generated based on a database of over 7 million images categorized into 205 possible place categories, trained using the Caffe framework [27]. While the 205 place categories it uses as outputs are chosen based on types of locations people are likely to encounter (including showers, basements), it represents a state of the art database and network that is continually updated and improved.

A concrete example of scene classification of the same location using the Places-CNN model [62] is shown for a satellite image of Eloy, AZ (Figure 3.1(a)) compared to an in-flight image from a parafoil system (Figure 3.1(b)) of the same region. Of 205 possible categories, the semantic classification output layer of the CNN for both images is broadly similar, despite different orientation, perspective, and lighting.

As described in Section 3.2, the possibilities afforded by this semantic classification are numerous, opening new mission modalities and capabilities in a range of challenging scenarios

(Section 1.2). Understanding imagery semantically allows vision to be better, rather than just more complicated, than traditional GPS navigation.

3.4 Localization and Landing: General Probabilistic Framework

In an airdrop context, the primary goal is to navigate towards (and land at) the optimal landing site. The goal of global localization is tightly coupled with solving the primary landing site problem. When the landing site is specified in global coordinates, localization is the entirety of the problem. In circumstances where the landing site is not specified explicitly, another metric must be defined. Defining optimal landing site(s) is dependent on where agents are located. With vision sensing, the same data is used for localization and landing site classification. As a result, an appropriate probabilistic framework to incorporate both objectives is potentially useful.

In the following sections, a framework for defining the problem is laid out in multiple pieces. In practice, it turns out considering the joint distribution of location and landing site is not needed in the cases considered. It remains instructive, however, to see how they can still be combined probabilistically naturally. Simplifications of these general forms will be used in more concrete ways in Chapters 5 and 6.

We can define the following random variables and distributions (in a global frame):

$$
\begin{aligned}
X &\equiv \text{current agent location} \\
Y &\equiv \text{optimal landing location} \\
Z &\equiv \text{data measurements, e.g. imagery} \\
P(x, y) &\equiv \text{joint distribution of current location and optimal landing location} \\
P(x, y|z) &\equiv \text{given measurements (vision+classification+other)}
\end{aligned}
$$

In general there are $K \geq 1$ agents, each with their own measurements z_k. Each has a motion

model and a communication protocol, and each agent estimates their own current location x_k and optimal landing location y_k within that environment at each time step t. Stated formally, each agent is estimating:

$$\hat{X}_k(t), \hat{Y}_k(t) = \underset{X_k(t), Y_k(t)}{\arg\max} P(X_k(t), Y_k(t)|Z) \tag{3.1}$$

This formulation is a modification of the conventional SLAM problem, where there is no landing site estimation, i.e. no Y term.

3.4.1 Environment Classification: Sets and Goodness

The concept of classifying locations by "sets" S is introduced to capture the perceptual aliasing present in many real world environments, enabling us to efficiently generate a probability distribution over all possible locations. These sets might group locations as being of $S \in \{$"road", "field", "forest", "lake", ...$\}$, or some combination of ML attributes, capturing the idea that locations are not visually unique, but may be grouped into unique sets. Formally, we can define these sets as follows:

$$\{S_m\}_{m=1}^{M}, S_m \cap S_l = \{\varnothing\} \ \ \forall S, m \neq l \tag{3.2}$$

where M is the number of sets for a particular environment. These sets can also be thought of as the result of a machine learning type classification. The utility of these sets is in simplifying the problem of matching observed and mapped data. Moreover, in a multi-agent context, this approach should improve localization in challenging, perceptual aliased, environments through the use of geometric constraints between agents (detailed in Section 3.6.1).

Based on the CNN output described in Section 3.3, the advantage of using sets over simply using the probability distribution of semantic descriptors is less pronounced, but for the clarity of presentation the set concept is helpful nonetheless for fixing ideas. The concept remains identical: locations are not visually unique, but may be classified into distinct cat-

egories. How this concept is applied with a CNN-generated semantic descriptor is described in Section 5.4.

3.4.2 Sensor Model and Filter

Assume a vision sensor outputs the set of the current observation correctly with some known probability distribution. In the notation defined above, let the probability of a sensor classifying environment measurements z_k for agent k correctly be as follows:

$$P(S(z_k) = S_m | S(x_k) = S_m) \equiv p \tag{3.3}$$

By using the chain rule it can be shown that Bayes' rule for joint distributions is exactly what one would naively expect:

$$P(x_k, y_k | z_k) = \frac{P(z_k | x_k, y_k) P(x_k, y_k)}{P(z_k)} \tag{3.4}$$

and measurement likelihood can similarly be defined:

$$P(z_k | x_k, y_k) = \frac{P(x_k | y_k, z_k) P(y_k | z_k) P(z_k)}{P(x_k | y_k) P(y_k)} \tag{3.5}$$

We can then state the form a Bayesian filter would take. The update equation would be:

$$P(\hat{x}_k(t), \hat{y}_k(t) | z_k(t)) = \frac{P(z_k(t) | x_k(t), y_k(t)) P(\hat{x}_k(t), \hat{y}_k(t) | z_k(t-1))}{\Sigma_x P(z_k(t) | x_k(t), y_k(t)) P(\hat{x}_k(t), \hat{y}_k(t) | z_k(t-1))} \tag{3.6}$$

And, assuming a motion model is available and reasonable assumptions about landing sites remaining suitable, the prediction equation for agent k operating alone would be:

$$P(\hat{x}_k(t+1), \hat{y}_k(t+1) | z_k(t)) = \frac{P(\hat{x}_k(t+1), \hat{y}_k(t+1) | \hat{x}_k(t), \hat{y}_k(t)) P(\hat{x}_k(t), \hat{y}_k(t) | z_k(t))}{\Sigma_x P(\hat{x}_k(t+1), \hat{y}_k(t+1) | \hat{x}_k(t), \hat{y}_k(t)) P(\hat{x}_k(t), \hat{y}_k(t) | z_k(t))} \tag{3.7}$$

These expressions serve to demonstrate that probabilistically, solving the localization problem together with the landing problem can be naturally defined. The first term of the right hand side of (3.7) $P(\hat{x}_k(t+1), \hat{y}_k(t+1) | \hat{x}_k(t), \hat{y}_k(t))$ is a Markov motion model, defined explicitly in Section 3.6.1, which propagates the state forward based on an agent's

possible next moves. Combined with the sensor model in expression (3.3) which gives us $P(z_k(t)|x_k(t), y_k(t))$, we have everything needed to run a Bayesian estimator for the joint distribution of location and landing site based on observed images $z_k(t)$.

This general form enables examination of how landing site and localization can be considered together. As noted in Section 3.4, however, when a satellite map is available and reliable, landing site quality can be calculated off-line in advance, and solving the localization problem completely solves the landing site problem as well. In other words, $P(y_k|x_k) = 1$, since effectively $Y_k = X_k$. On the other hand, when no map prior is available, there is no means for estimating X_k whatsoever, thus $P(y_k|x_k) = P(y_k)$ since $P(x_k, y_k) = P(x_k)P(y_k)$.

The scenario where a map prior is available but may be unreliable could be where using the full joint distribution approach derived here would be most useful. Practically, however, if an agent is having difficulty localizing in a map, it would likely conclude that its map prior is unreliable, and would then need to revert to landing site identification based solely on its observed imagery.

As a result, Chapter 5 explores the scenario where localization is possible with a map prior, and Chapter 6 considers landing site identification when no map prior is available.

3.5 Landing Site Optimality

In choosing landing sites in a GPS-denied setting, the approach presented here assumes that global localization may not always be possible or practical. As a result, visually classifying landing sites is necessary as a complement, or potential replacement, to traditional, hard-coded, global landing coordinates. Based on the introduction of semantic scene understanding (Section 3.3) and set classification and sensing (Sections 3.4.1, 3.4.2), a natural form for landing site optimality can be defined.

We begin by defining what a globally optimal landing site cost function would look like. In practice, we will only be able to explicitly calculate some of the terms we define given the assumed knowledge and communications limitations, but it remains a helpful construct for fixing ideas nonetheless.

Let $G \in [0, 1]$ be defined as normalized landing optimality, where higher values correspond to better landing locations. G is defined based on a number of criteria, including intrinsic landing quality of a location, quality of surrounding locations, and some notion of distance of the agent(s) from that location. These criteria can be combined into an objective function and optimized to find the best landing site. To fix ideas, a choice of objective function could be:

$$G(g_y, \breve{g}_y, \mathcal{D}_y) = w_1 g_y + w_2 \breve{g}_y + w_3 \mathcal{D}_y \tag{3.8}$$

where

$$g_y \in (0, 1) \equiv \text{intrinsic quality of location based on its set}$$
$$\breve{g}_y \in (0, 1) \equiv \text{quality of location surroundings}$$
$$\mathcal{D}_y \in (0, 1) \equiv \text{notion of location distance from } k \text{ agents}$$
$$\Sigma w_i = 1,\ w_i \equiv \text{non-negative scalar weightings}$$

The intuition for this objective function is straightforward. By increasing w_1, w_2, agents favor objectively better locations, whereas by increasing w_3 areas more easily reached by the group are favored. This cost function can be used to update the relative optimality of landing locations y_k based on observations z_k, and using a Bayesian update framework, for instance, Y_k will remain a valid probability distribution.

In defining $\mathcal{D}_y$, one possibility is:

$$\mathcal{D}_y = \frac{1}{k}\Sigma_k \left(\frac{\mathrm{d}_{k,\mathrm{go}}}{\mathrm{d}(y - \hat{x}_k) + \mathrm{d}_{k,\mathrm{go}}} \right) \tag{3.9}$$

where $\mathrm{d}(y - \hat{x}_k)$ corresponds to the distance of the considered landing location y from agent k's estimated location $\hat{x}_k$, and $\mathrm{d}_{k,\mathrm{go}}$ is the remaining horizontal distance agent k can travel, estimated based on current altitude and sink rate. For parafoil systems, the total amount of horizontal distance remaining in flight can be roughly estimated by a known mean sink rate [13]. This choice of $\mathcal{D}_y$ maintains the convention of larger values corresponding to better locations (meaning shorter relative distances for all agents to travel). With this definition, values of $\mathcal{D}_y < 1/2$ correspond to locations that cannot be reached by at least half of all agents. Other choices of landing site optimality are certainly possible, but this represents a simple, concrete example.

For simplicity, define the measurement model as follows: a vision sensor provides the set of current observation correctly with some probability, which then immediately gives the inherent goodness g_y. Furthermore, by averaging the sets surrounding each location using map priors and measurements, $\breve{g}_y$ can be estimated as well. Stated formally, as in Section 3.4.2, the assumed sensor model can be categorized as follows:

$$P(G(z_k) = g_y | G(y_k) = g_y) \equiv p \tag{3.10}$$

With this, or a similar definition in place, the landing optimality of an environment can be classified visually both in a map prior, as well as in real-time, especially if the only map available is outdated or inaccurate. This enables a single agent to find a landing site on its own in a GPS-denied setting.

In practice, calculating the second two terms in expression (3.8) throughout flight may not be possible. To begin with, assessing the average values $\breve{g}_y$ of surrounding areas requires more information than is available early on in flight. Furthermore, calculating $\mathcal{D}_y$ can be

done instantaneously by an agent for its current location if it knows the distance to all other agents, but the calculation is immediately outdated as all the agents continue moving within the environment.

As a result, while (3.8) is a helpful construct for defining optimal landing sites, it may be a calculation only made periodically. Thus $w_2 = w_3 = 0$ for the majority of flight, hence the first term, g_y, the intrinsic quality of an observed location, will be the primary topic explored further. Focusing on the group calculation of g_y will enable investigation of how multiple agents classifying their respective locales and communicating rudimentary information can still greatly enhance landing site identification (Chapter 6).

3.6 Multi-agent Estimation: Choice of Environment Representation

The concept of semantically understanding imagery probabilistically for navigation has not been widely explored for good reason. Without additional constraints, unique localization is nearly impossible, since perceptual aliasing is unavoidable, by design. Odometry is one tool to constrain the multiplicities of semantically similar locations, though this may require a substantial amount of time to move through an environment, and will not help in certain scenarios (i.e. middle of a large, homogenous, area). Multi-agent collective localization can provide a key missing piece, not as a simply more complicated version of single-agent navigation, but as a crucial component for showing the power of semantic visual classification.

3.6.1 Discrete Distribution Approach: Grid

Thus far, the random variables and distributions defining the problem (Section 3.4) have been left in general form. In practice, the choice of a particular representation scheme is one of the more challenging aspects of navigation and estimation, given the limitations of back-end solvers (Section 2.4) to effectively and efficiently produce usable algorithms. The choice of Gaussian structure is common and convenient. However, given the compressed sensing model described of Section 3.4.2, for maximum flexibility, using a discrete distribution to

represent space is a reasonable approach, as done in FAB-MAP [18]. A key difference between this approach and FAB-MAP, however, is the inherently branched map of 2D space, as opposed to a linear path along city streets.

A concrete environment and motion model is now introduced, but the algorithms have more general applicability than to the specific model considered here. For clarity of presentation, only the localization part of the problem is described.

Each agent begins in a random location $x_k(0)$ at $t = 0$ within the environment and takes a random walk for t time steps. To fix ideas, let us assume the locations described form a grid of blocks (b_i), as in Figure 3.2(a), and at every time step the agent uniformly may make at most 8 moves to the adjacent locations, or remain in its current location block.

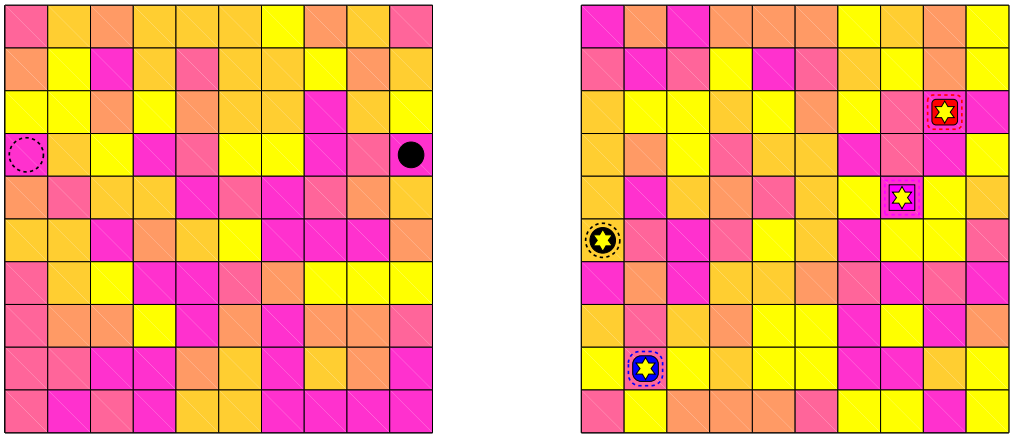
Actual parafoil dynamics are more consistent with unicycle type motion, which can be captured by adding heading states and constraints to this environment. This will be pursued in Section 5.5.

This motion model can be compactly captured in a Markov model:

$$x_k(t+1) = Tx_k(t) \tag{3.11}$$

where T is a left stochastic matrix. This defines the probability distribution $P(x_k(t+1)|x_k(t))$ for each agent k in expression (3.7) in Section 3.4.2.

For simplicity, we assume that all agents can broadcast and receive transmissions from each other agent every time step. This assumption can be relaxed in the future, but allows for seeing an upper bound on the utility of a multi-agent approach. Each agent transmits its own estimate of being located in each set $P(S(\hat{x}_k))$, where the probability of each set is the



(a) (b)

Figure 3·2: Representative map environment with 5 different location sets, represented by different colors. (a) A single agent, even with perfect measurement of current set, may estimate its position far from its actual location. (b) Multiple agents, each communicating their own set probability (and implicitly scalar distance), can estimate position significantly more accurately (correct estimates indicated by star).

sum of its corresponding location blocks b_i:

$$P(S(\hat{x}_k) = S_m) = \Sigma_{i \in S_m} P(\hat{x}_k(t) = b_i), \forall m \in M. \tag{3.12}$$

This reduces the amount of data transmitted, sending only the distribution of sets, rather than the entire $\hat{x}_k$ distribution. In addition, each agent is assumed to know the corresponding scalar distance to each other agent $\mathrm{d}(x_k, x_h)\ \forall\ h \in K$, consistent with measuring signal strength of messages received (RF-ranging), for instance. This measurement is assumed to have bounded noise ε which must be less than the smallest value the distance function $\mathrm{d}(.,.)$ takes for adjacent locations in this particular environment (i.e. the scalar measurements give unambiguous geometric constraints).

The maximum a posteriori probability (MAP) of $P(\hat{x}_k|z_k)$ is chosen to define the current estimated block $\tilde{x}_k$, corresponding to the block b_i with the highest probability. A simple

algorithm to incorporate information from other agents is to use the measured distances between each agent $d_{k,h} = \mathrm{d}(x_k, x_h)$ and set estimates $S(\hat{x}_h)\ \forall\ h \in K$ from each other agent to pairwise re-weight the probability estimate of each location block.

For each agent k, weight each location which is $d_{k,h}$ distance away to agent h by the probability agent h is in that location, estimated using the transmitted set probabilities from agent h. To find the consistent matches, use a precomputed lookup table (distTable) from the map of the environment which has distances between locations of each set to one another. Recalling $S_m(\hat{x}_h)$ is the set for a particular location, this gives:

$$P(x_k = b_i | S(\hat{x}_h), d_{k,h}) = \begin{cases} \frac{S_m(\hat{x}_h)}{|S_m|} & : |\text{distTable - } d_{k,h}| \leq \varepsilon \\ 0 & : |\text{distTable - } d_{k,h}| > \varepsilon \end{cases} \tag{3.13}$$

Thus the final multi-agent update step, modified from (3.6) can be written as:

$$P(\hat{x}_k | z_k, S(\hat{x}_h), d_{k,h}) = \frac{P(z_k|x_k)P(x_k|S(\hat{x}_h), d_{k,h})P(\hat{x}_k|z_k(t-1))}{\Sigma_x P(z_k|x_k)P(x_k|S(\hat{x}_h), d_{k,h})P(\hat{x}_k|z_k(t-1))} \tag{3.14}$$

where the measurements from other agents now appear explicitly. This could be performed for each agent $h \in K \neq k$ as shown, or performed in batch. The predict step (3.7) remains unchanged. This procedure is summarized in Algorithm 1:

Algorithm 1 Multi-Agent Semantic Sensing Optical Localization (MASSOL)

for all b_i **do**
  for all $h \in K \neq k$ **do**
    $d_{k,h} \leftarrow \mathrm{d}(x_k, x_h)$
    $P(x_k = b_i | S(\hat{x}_h), d_{k,h}) \leftarrow 0$
    for all $S_m, m \in M$ **do**
      matches $\leftarrow$ find($|\text{distTable - } d_{k,h}| \leq \varepsilon$)
      for all matches **do**
        $P(x_k = b_i | S(\hat{x}_h), d_{k,h})$ += $\frac{S_m(\hat{x}_h)}{|S_m|}$
      end for
    end for
  end for
  $P(x_k = b_i | z_k, S(\hat{x}_h), d_{k,h}) \leftarrow P(z_k|x_k)P(x_k = b_i|S(\hat{x}_h), d_{k,h})P(\hat{x}_k|z_k(t-1))$
end for
$P(\hat{x}_k | z_k, S(\hat{x}_h), d_{k,h}) \leftarrow \frac{P(\hat{x}_k|z_k, S(\hat{x}_h), d_{k,h})}{\Sigma_x P(z_k|x_k)P(x_k|S(\hat{x}_h), d_{k,h})P(\hat{x}_k|z_k(t-1))}$

3.6.2 Example Implementation

The model described in Sections 3.4-3.6 is implemented in MATLAB, and simulations are conducted for a 10x10 block environment (100 possible locations) classified using 5 sets, distributed randomly according to a uniform distribution. Three different sensor models are considered, with $p = \{0.6, 0.8, 1\}$ (as defined in (3.3), and scenarios with 1 through 4 robots are simulated for each. The results for correctly estimated and normalized average distance from correct location averages are shown in Figures 3·3 and 3·4, respectively. Representative environment plots are shown in Figure 3·2. In general, the benefit of multiple agents in improving localization is clear.

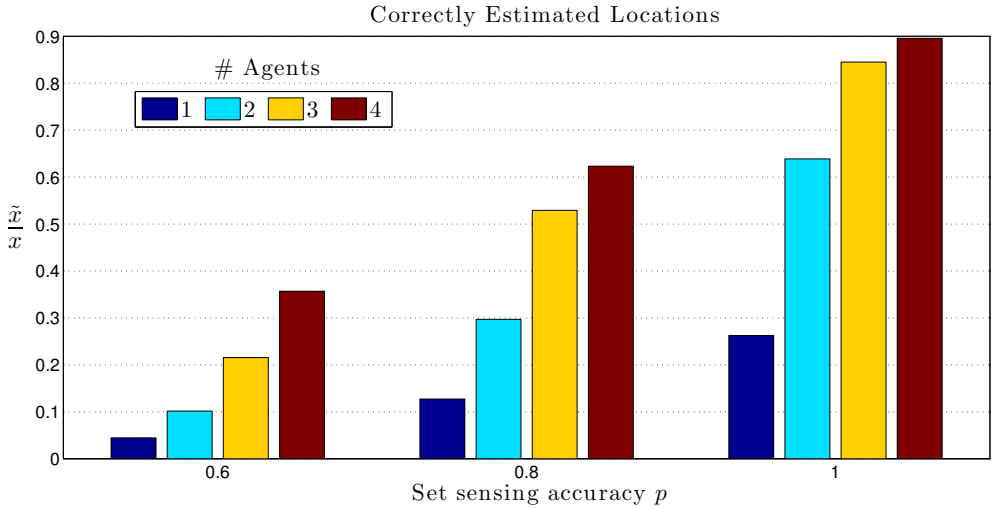


Figure 3·3: Fraction of accurately estimated locations $\left(\frac{\hat{x}}{x}\right)$ as a function of sensor model (3.3), where p indicates with what accuracy sensor returns the correct set of current location (higher is better). Higher sensor fidelity increases accuracy across the board. Increasing the number of agents significantly improves estimation accuracy. With 4 agents, sensing accuracy of only 0.6 is more accurate than perfect estimation (1) with a single agent. This Figure (and Figure 3·4) are generated for a 10x10 environment (100 possible locations) classified using 5 sets, distributed randomly according to a uniform distribution. For each sensor and number of agents combination, averages are shown for 10 simulations of 100 time steps each.

A natural question arises concerning the existence of edges in the environment. Namely, one

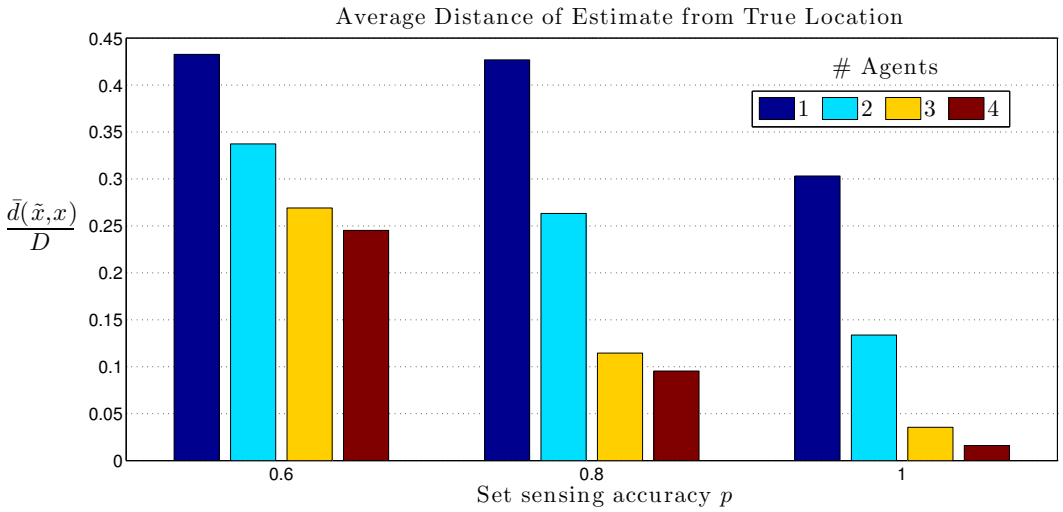


Figure 3·4: Normalized average distance $\left(\frac{\bar{d}(\hat{x},x)}{D=\max(d_{ij})}\right)$ of estimate from actual position as a function of sensor model , where p indicates with what accuracy sensor returns the correct set of current location (lower is better). This shows how far off, on average, the estimation is in each of the scenarios described in Figure 3·3. Increased sensor accuracy improves all cases. As in Figure 3·3, 4 agents using degraded sensing ($p = 0.6$) still, on average, estimate their locations better than a single agent with perfect sensing (1). This shows that not only do more agents estimate their exact location better than a single system with perfect sensing, they also estimate the general area of their locations more accurately.

might wonder if the multi-agent success results primarily from agents at opposite edges or corners immediately constraining their distributions, rather than actually using the sensor data more effectively. In other words, are the geometric constraints afforded by multiple agents only effective in a map with edges? To isolate edge effects, another series of simluations is conducted. The agents are now constrained to move within a 10x10 environment that is embedded in a larger 30x30 environment. This ensures that even if two agents are at opposite corners, the distribution will not collapse by virtue of geometry alone.

To confirm the absence of edge effects with this setup, tests are first run with 1 set–a completely homogeneous environment. As expected, the addition of multiple agents has no effect in this case, since the agents have no information available from the environment to constrain the initial uniform distribution.

In the 30x30 environment there are 900 possible locations instead of 100 as in the 10x10; thus the number of sets in this environment is tested both with 5 sets (as in the previous simulations), as well as 10 sets. A sample image from these simulations is shown in Figure 3·5, and results plots are given in Figures 3·6 and 3·7.

The advantage of multiple agents remains clear, indicating the localization algorithm is not dependent on the effects of edges alone to function as designed. Going forward, we can now ignore the presence of edge effects, since they are demonstrably not the primary source of multi-agent performance improvements.

These example simulations show the combined approach of semantic scene understanding along with multi-agent constraints is a reasonable method to address sharing of estimates without overlap (assuming a map prior), as well as perceptual aliasing. The results confirm the effectiveness of simple multi-agent communication for greatly improving accuracy of localization estimation. In the demonstration environments the sets are independently distributed, whereas in realistic environments they would often be distributed in a correlated way, as will be explored in later chapters. Other constraints, such as odometry, dropping communication channels, and maximum communication distance between agents could also be incorporated into this model, which will remain effective, given its pairwise nature.

Chapter 4 will demonstrate some mathematical verification for these results, before application to in-flight camera imagery is pursued in Chapters 5 and 6.

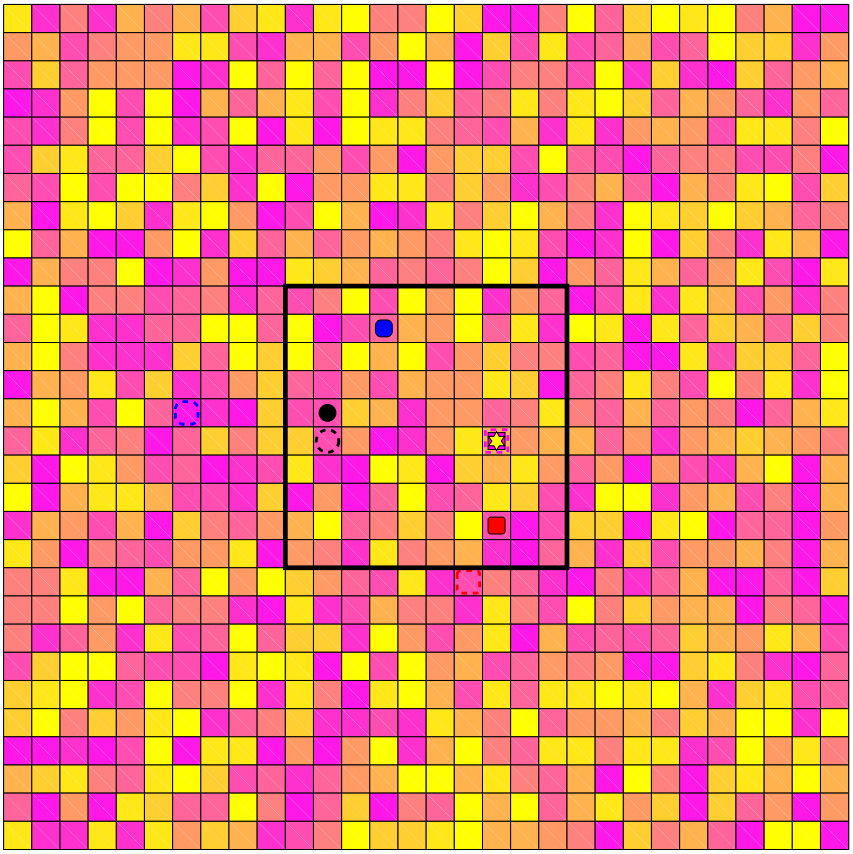


Figure 3·5: 30x30 grid environment to isolate effects of edges. Agents can only move with the inner 10x10 box, but are estimating location within the entire map. Given the proliferation of locations (900 vs 100 in previous simulations) the number of distinct sets is tested with both 5 and 10 sets.

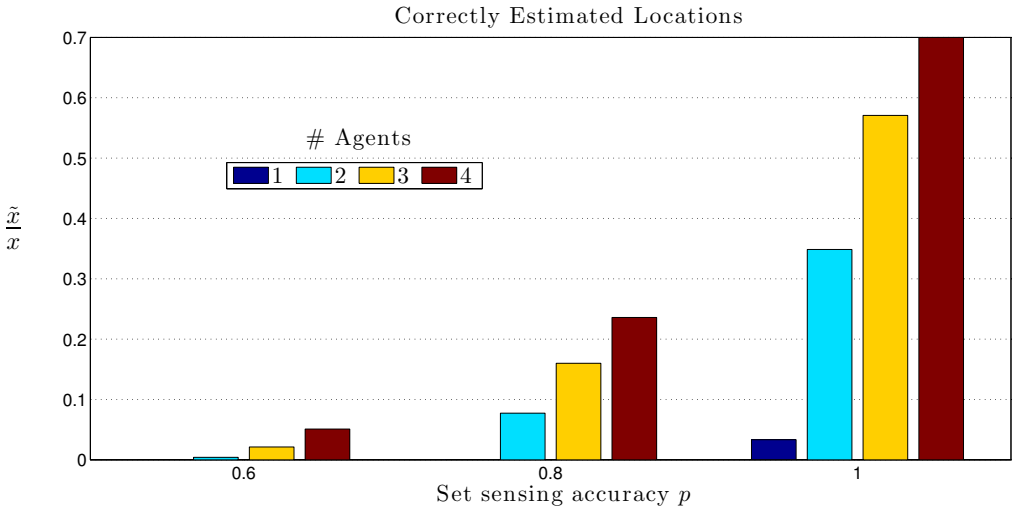


Figure 3·6: Fraction of accurately estimated locations $\left(\frac{\hat{x}}{x}\right)$ as a function of sensor model for scenario with no edges in a 30x30 environment uniformly classified into 5 sets (averages from 10 runs of 100 steps each shown). As expected, accuracy degrades substantially from the results shown in Figure 3·3. The trend of multiple agents being beneficial, however, remains clear, demonstrating that edge effects are not the primary source of multi-agent performance improvements.

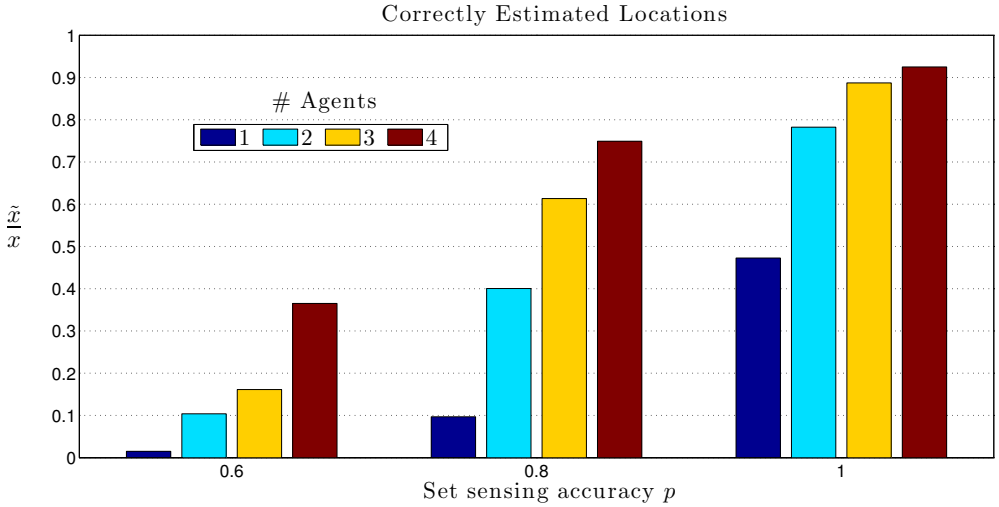


Figure 3·7: Fraction of accurately estimated locations $\left(\frac{\hat{x}}{x}\right)$ as a function of sensor model for scenario with no edges in a 30x30 environment uniformly classified into 10 sets (averages from 10 runs of 100 steps each shown). These results show how much of the performance decline shown in Figure 3·6 results primarily from the vastly larger number of locations in each set (even with 10, each set contains 90 locations, versus 20 in the case of the 10x10 grid discretized into 5 sets), as opposed to the loss of edge effects.

Chapter 4

Mathematical Foundation

While the simulation results in Section 3.6.2 are compelling, having a stronger mathematical foundation for the multi-agent coordination algorithm is instructive. The following Chapter will show that, in the environment with set sensing model developed in Chapter 3, multiple agents are guaranteed to have the same or less uncertainty when compared against a single agent working alone when all agents have perfect sensing capabilities. Moreover, a concrete example will demonstrate how multiple agents working together with degraded sensing can reduce uncertainty better than a single agent working alone with perfect sensing, providing some helpful intuition.

4.1 Multi-agent Terminology

We now introduce additional terminology based on the development in Chapter 3 to expand the details of Algorithm 1. As in Section 3.6.1, the discrete environment is divided into blocks (locations) b, each of which is grouped into sets S_m. There are M sets, each of which contains $|S_m|$ blocks. In general there are K agents, which can measure the scalar distance between each other, $d_{k,h}$ being the distance between agent k and h. Let:

$$b^k_{S^i_m}(t) \equiv \text{Probability of block } i \text{ in set } S_m \text{ for agent } k \text{ at time } t \tag{4.1}$$

For simplicity we will only include the time t when needed, since we will focus exclusively on first step analysis here.

As in section 3.4.2, the vision sensor observes set S_l of image z_k, which will match the true set of that image S_m with some probability p, and incorrectly classify the set uniformly

among the remaining $M-1$ sets. Based on equation (3.3) we define a function f to capture this:

$$f(S_l, S_m) \equiv \begin{cases} P(S_l = S_m|S_m) \equiv p \\ P(S_l \neq S_m|S_m) \equiv \frac{1-p}{|M-1|} \quad \forall S_l \in M \setminus S_m \end{cases} \tag{4.2}$$

This function is the local vision measurement each agent makes on its own. This is equivalent to defining an M dimensional confusion matrix where p are the terms on the diagonal and $\frac{1-p}{|M-1|}$ are the off-diagonal terms.

Each agent k broadcasts the distribution of its location estimate as a function of sets $P^k_{S_m}$, where

$$P^k_{S_m} = \sum_{i \in |S_m|} b^k_{S^i_m} \tag{4.3}$$

This is equivalent to expression (3.12), but stated in more concise notation. In the single agent case, where each block in each set is equiprobable, this expression is equivalent to (4.2) when no prior information is available.

The core concept of how the multi-agent algorithm reduces uncertainty is based on constraining the distributions with scalar inter-agent distances of agents k and h. We use the notation $\sum_{j \in S_l : d_{k,h}}$ to express that.

With this notation, we are now ready to restate the main portion of Algorithm 1. Taking the definition in expression (4.1):

$$b^k_{S^i_m}(t) = b^k_{S^i_m}(t-1) \sum_{h \in K \setminus k} \sum_{l \in M} \sum_{j \in S_l : d_{k,h}} \frac{f(S_l, S_m)}{|S_l|} b^h_{S^j_l} \tag{4.4}$$

Restated with explanation of each term:

$$b^k_{S^i_m}(t) = \underbrace{b^k_{S^i_m}(t-1)}_{\text{prior}} \underbrace{\sum_{h \in K \setminus k}}_{\text{Agents}} \underbrace{\sum_{l \in M}}_{\text{Sets}} \underbrace{\sum_{j \in S_l : d_{k,h}}}_{\text{Distance constraint}} \underbrace{\frac{f(S_l, S_m)}{|S_l|}}_{\text{Vision sensor}} \underbrace{b^h_{S^j_l}}_{\text{Prob of agent } h}$$

This defines how the distribution for each block in each set is calculated based on incorporation of the prior value, its own vision sensing, and distance constraints with other agents. For a given measurement between agents k and h, the distance constraint $d_{k,h}$ is constant. The normalization step to ensure it remains a valid distribution (summing to 1) is handled in batch by division by the sum of total $b^k_{S^i_m}$ blocks.

When a single agent is working alone, the expression collapses, since only the prior and instantaneous vision measurement is available:

$$b_{S^i_m}(t) = b^k_{S^i_m}(t-1)\frac{f(S_l, S_m)}{|S_l|} \tag{4.5}$$

In practice, the prior term $b^k_{S^i_m}(t-1)$ begins as uniformly distributed among the elements B of the environment, hence takes value $\frac{P^k_{S_m}}{|S_m|}$ at the first step.

We now briefly introduce the concept of entropy, which will allow us to make some statements about how this multi-agent localization approach compares to the single agent case.

4.2 Entropy

There are numerous ways to describe the uncertainty in a probability distribution, and to compare one distribution to another. In discrete distributions, the concept of entropy from information theory is particularly useful. Defined originally by Shannon in his seminal work which effectively created the field of information theory (originally published in 1948 in [51] [52], reprinted in [53]), entropy can be stated in numerous equivalent ways. For our

purposes, the following definition will be sufficient:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{4.6}$$

where $|\mathcal{X}|$ defines the number of elements in the range space of X. The choice of base for the log can be altered depending on the desired interpretation of the value. We will work in $\log_2$ consistent with Shannon's approach concerning the bits of data needed to represent a distribution (thus, heretofore $\log \equiv \log_2$). The raw value will not be of particular interest here, but rather the relative values of different distributions.

In practice, the negative sign is often brought into the log term giving

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} \tag{4.7}$$

Entropy has a number of helpful properties that will be useful [16].

$$H(X) \geq 0 \tag{4.8}$$

$$H(X|Y) \leq H(X) \tag{4.9}$$

$$H(X) \leq \log |\mathcal{X}| \tag{4.10}$$

$$H(p) \text{ is concave in } p \tag{4.11}$$

In words, these properties translate to: non-negativity of entropy, conditioning reduces entropy, uniform distribution maximizes entropy, and the further from uniform a distribution is the lower its entropy function will be. In addition, entropy has a grouping property:

$$\begin{aligned} H(p_1, p_2, ..., p_m) = {} & H(p_1 + p_2 + ... + p_k, p_{k+1} + ... + p_m) \\ & + (p_1 + p_2 + ... + p_k) H\left(\frac{p_1}{p_1 + p_2 + ... + p_k}, ..., \frac{p_k}{p_1 + p_2 + ... + p_k}\right) \\ & + (p_{k+1} + ... + p_m) H\left(\frac{p_{k+1}}{p_{k+1} + ... + p_m}, ..., \frac{p_m}{p_{k+1} + ... + p_m}\right) \end{aligned} \tag{4.12}$$

Together with the notation defined in Section 4.1, we can make some statements about how

multiple agent localizing with Algorithm 1 compare to the single agent case.

4.3 Comparing Single and Multi-agent Entropy

The goal of this section is to demonstrate that multiple agents working together following Algorithm 1 are guaranteed to have lower entropy in their respective distributions than a single agent, when all agents have identical, perfect set-sensing capabilities.

In the environments defined here we have allowed for variable size sets. In the most challenging environments, however, not only would the probabilities within sets be uniform, but the sets themselves would also be uniformly distributed in the environment, meaning they have the same number of elements. This corresponds to the definition of visual richness (VR) defined in Section 3.2.1. Hence we will let:

$$|S_1| = |S_2| = ... = |S_M| \equiv |S| \tag{4.13}$$

For clarity we will retain the different set notations where appropriate.

We begin by writing down the entropy for the single agent case. Combining expressions (4.5) and (4.3) with (4.12) gives:

$$\begin{aligned} H(b_{S_1^1}, ..., b_{S_1^{|S_1|}}, b_{S_2^1}, ..., b_{S_M^{|S_M|}}) = {} & H(P_{S_1}, P_{S_2}, ..., P_{S_M}) \\ & + P_{S_1} H\left(\frac{b_{S_1^1}}{P_{S_1}}, ..., \frac{b_{S_1^{|S_1|}}}{P_{S_1}}\right) \\ & + ... \\ & + P_{S_M} H\left(\frac{b_{S_M^1}}{P_{S_M}}, ..., \frac{b_{S_M^{|S_M|}}}{P_{S_M}}\right) \end{aligned} \tag{4.14}$$

Based on the definition of $b_{S_m^i}$, we know that within each set, each element is equiprobable, which is the definition of a uniform distribution. From expression (4.10) we can then simplify

(4.14):

$$H(b_{S_1^1},...,b_{S_1^{|S_1|}},b_{S_2^1},...,b_{S_M^{|S_M|}}) = H(P_{S_1},P_{S_2},...,P_{S_M}) + P_{S_1}\log|S_1| + ... + P_{S_M}\log|S_M| \tag{4.15}$$

Using expression (4.13), and recalling $\Sigma P_{S_m} = 1$ we can further simplify (4.15) to:

$$H(b_{S_1^1},...,b_{S_1^{|S_1|}},b_{S_2^1},...,b_{S_M^{|S_M|}}) = H(P_{S_1},P_{S_2},...,P_{S_M}) + \log|S| \tag{4.16}$$

This expression reveals some important intuition about the single-agent case. The first term on the right hand side corresponds to the entropy associated with the distribution of sets, while the second term relates to the entropy of the locations within each set. Before any measurement, where the sets themselves are equiprobable, expression (4.16) would become:

$$H(b_{S_1^1},...,b_{S_1^{|S_1|}},b_{S_2^1},...,b_{S_M^{|S_M|}}) = \log|M| + \log|S| = \log|B| \tag{4.17}$$

where $|M|$ is the number of sets, $|S|$ is the number of elements in each set, and $B = |M||S|$ is the total number of blocks in the environment. This is exactly what we should get for the case where there are B uniformly distributed locations.

Of particular interest is the case where $P_{S_m} = 1$, i.e. in the sensor model expression (4.2) $p = 1$ (the agent has perfect sensing of the set of its current location). This is intuitively a bound on how well a single agent could do operating on its own. The entropy relating to the distributions of sets becomes zero (since the set is known), hence we can immediately conclude from expression (4.17):

$$H(b_{S_1^1},...,b_{S_1^{|S_1|}},b_{S_2^1},...,b_{S_M^{|S_M|}}) = \log|S| \tag{4.18}$$

Where all of the uncertainty (and entropy) comes from the unknown distribution within the observed set. This then allows us to state the following:

Lemma 4.3.1. *The lower bound on the entropy function of a single agent operating alone in the environment defined here is* $\log|S|$. *A single agent cannot perform any better than*

in the case of $p = 1$ *in (4.2), when it achieves entropy of* $\log|S|$.

Turning to the multi-agent case, the original expression is similar, but now carries the k superscript for agent k since we are using expression (4.4)

$$\begin{aligned} H(b^k_{S^1_1},...,b^k_{S^{|S_1|}_1},b^k_{S^1_2},...,b^k_{S^{|S_M|}_M}) &= H(P^k_{S_1},P^k_{S_2},...,P^k_{S_M}) \\ &+ P^k_{S_1} H\left(\frac{b^k_{S^1_1}}{P^k_{S_1}},...,\frac{b^k_{S^{|S_1|}_1}}{P^k_{S_1}}\right) \\ &+ ... \\ &+ P^k_{S_M} H\left(\frac{b^k_{S^1_M}}{P^k_{S_M}},...,\frac{b^k_{S^{|S_M|}_M}}{P^k_{S_M}}\right) \end{aligned} \tag{4.19}$$

As in the single agent case, if we assume the agents have perfect sensing of their respective sets, the first term in this expression becomes zero. Hence, in the case where $p = 1$, where p is as defined in expression (4.2), the entropy for multiple agents is:

$$\begin{aligned} H(b^k_{S^1_1},...,b^k_{S^{|S_1|}_1},b^k_{S^1_2},...,b^k_{S^{|S_M|}_M}) &= P^k_{S_m} H\left(\frac{b^k_{S^1_m}}{P^k_{S_m}},...,\frac{b^k_{S^{|S_m|}_m}}{P^k_{S_M}}\right) \\ &= H\left(\frac{b^k_{S^1_m}}{P^k_{S_m}},...,\frac{b^k_{S^{|S_m|}_m}}{P^k_{S_M}}\right) \end{aligned} \tag{4.20}$$

where we know $P^k_{S_m} = 1$ by virtue of perfect set sensing.

Recalling that $H(X) \leq \log|\mathcal{X}|$, we thus know that

$$H\left(\frac{b^k_{S^1_m}}{P^k_{S_m}},...,\frac{b^k_{S^{|S_m|}_m}}{P^k_{S_m}}\right) \leq \log|S_m| \ \forall m \in M \tag{4.21}$$

Comparing expressions (4.20) with (4.21) we can then state our result.

Theorem 4.3.2. *In an environment categorized into discrete sets* S_m *(3.2), where set elements* $b^k_{S^i_m}$ *are defined over a metric space, and agents* $k \in K$ *have a vision sensor model (4.2) with equal probability of identifying sets correctly* p*, multiple agents working together following protocol (4.4), i.e. Algorithm 1, are guaranteed to each have distributions with entropy less than or equal to the distribution entropy of a single agent working alone when*

$p = 1$. *Furthermore, a single agent cannot perform any better than in the case of* $p = 1$. *Hence multiple agents working together with* $p = 1$ *are guaranteed to reduce entropy as much, if not more, than the best a single agent can achieve alone.*

Proof. Using the grouping (4.12), uniform maximum (4.10), and concavity (4.11) properties of entropy, we can write expressions (4.21), which allows us to demonstrate inequality in expressions (4.18) and (4.20) term by term. Hence, we can conclude:

$$H(b^k_{S^1_1}, ..., b^k_{S^{|S_1|}_1}, b^k_{S^1_2}, ..., b^k_{S^{|S_M|}_M}) \leq H(b_{S^1_1}, ..., b_{S^{|S_1|}_1}, b_{S^1_2}, ..., b_{S^{|S_M|}_M}) \tag{4.22}$$

The second statement follows directly from Lemma 4.3.1. □

Other than defining the environment in terms of sets with elements in a metric space, no other constraints on the environment are required. This confirms the results in Section 3.6.2, which demonstrated that edge effects are not required for Algorithm 1 to be effective.

There is an important subtlety to this result. While the conclusion of Theorem (4.3.2) is far from surprising, in that we would expect multiple agents working together to reduce uncertainty more than a single agent if all have the same local sensing abilities, when sensing degrades (i.e. probability $p < 1$) this will not always hold. The intuition for this follows from comparing expression (4.16) and (4.19). While we can establish strict inequality for entropy within sets from (4.21), the reweighting procedure could result in the between-set entropy increasing. Depending on the particular environment and measurement, that increase could be greater than the reduction in entropy within sets. In Section 4.4.2, we will construct a reasonable scenario where at certain probabilities p, multiple agents reduce entropy more than a single agent, while at other probability p values they will actually slightly increase entropy.

What's important to recall, however, is overall entropy is only one metric for determining the success of an algorithm. As will be shown in Section 4.4.2, while overall entropy may not strictly decrease with the multi-agent algorithm proposed here, the within-set entropy reduction enables accurate estimation of agent location even if overall entropy may increase.

In practice, based on the simulation results in Section 3.6.2, and as will be demonstrated in Chapter 5, multiple agents following Algorithm 1, even in cases of degraded sensing, perform substantially better than single agents. To be sure, in cases where some agents share misleading information, whether from incorrect measurements, or in an adversarial fashion [33], one would expect multiple agents to perform worse than a single agent. In particular circumstances, however, whether as a result of very low VR environments or structural factors within those environments, multiple agents, even cooperating properly and measuring their environment probabilistically accurately, can increase overall distribution entropy.

4.4 Multiple Agents Outperforming Single Agent with Sensing Disparity

Another helpful intuition from expression (4.17) is the independent weight the entropy from sets and blocks has for the single agent case. With no way to overcome the uncertainty within a set between individual blocks, all reduction in entropy comes by virtue of reducing the uncertainty between sets. This means that when comparing to a multi-agent approach that can reduce entropy among sets as well as among blocks within sets, we can intuitively see there will be scenarios where degraded multi-agent sensing is more capable than higher fidelity single-agent sensing, consistent with the results in Section 3.6.2.

As we would expect, this will depend on the relative sensing disparity, as well as the environment and constraints afforded by information from additional agents. To show how this is reflected mathematically, we will now proceed through a simplified example, where the relative impact of these effects will emerge naturally.

4.4.1 Simplified Sensing Environment Scenario

Let us take an environment of VR $= \frac{1}{2}$, which translates to being divided into two sets S_1 and S_2 of the same number of elements, $|S_1| = |S_2| \equiv |S|$. We again assume every agent has sensor model (4.2). We can immediately write down the entropy for a single agent based

on (4.16):

$$H(b_{S_1^1},...,b_{S_1^{|S_1|}},b_{S_2^1},...,b_{S_2^{|S_2|}}) = H(P_{S_1},P_{S_2}) + \log|S| = h_B(p) + \log|S| \tag{4.23}$$

where we have introduced the binary entropy function $h_B(p) = -(1-p)\log(1-p) - p\log(p)$. By the concavity property (4.11) and symmetry it should be clear that $h_B(p)$ is maximized at $p = (1-p) = \frac{1}{2}$, and for any given p, $h_B(p) = h_B(1-p)$. In our context, the case where $p = (1-p) = \frac{1}{2}$ would again correspond to the original uniform distribution where both sets are equally likely, as in (4.17).

The case of greatest interest is where a single agent has perfect sensing, i.e. probability $p = 1$, the scenario in which we demonstrated Theorem 4.3.2. Since $h_B(1) = h_B(0) = 0$, the entropy for a single agent case then simply becomes:

$$H(b_{S_1^1},...,b_{S_1^{|S_1|}},b_{S_2^1},...,b_{S_2^{|S_2|}}) = \log|S| \tag{4.24}$$

since the only uncertainty remaining is within the elements of the observed set.

For the multi-agent scenario, let us assume there are two agents k and h a distance $d_{k,h}$ away from each other, and each observes the environment For agent k, the entropy of its distribution is as follows:

$$\begin{aligned} H(b^k_{S_1^1},...,b^k_{S_1^{|S_1|}},b^k_{S_2^1},...,b^k_{S_2^{|S_2|}}) = {} & h_B(P^k_{S_1}) \\ & + P^k_{S_1} H\left(\frac{b^k_{S_1^1}}{P^k_{S_1}},...,\frac{b^k_{S_1^{|S_1|}}}{P^k_{S_1}}\right) \\ & + P^k_{S_2} H\left(\frac{b^k_{S_2^1}}{P^k_{S_2}},...,\frac{b^k_{S_2^{|S_2|}}}{P^k_{S_2}}\right) \end{aligned} \tag{4.25}$$

where we again can use the binary entropy function h_B and with only 2 sets we know $P^k_{S_1} = 1 - P^k_{S_2}$.

Let us assume a scenario where agent k observes S_1 and agent h observes S_2, both with probability p. From expression (4.4) we can then write down expressions for $b^k_{S_1^i}$ and $b^k_{S_2^i}$:

$$b^k_{S_1^i} = \frac{P^k_{S_1}}{|S_1|}\left(\sum_{j\in S_1:d_{k,h}} \frac{p}{|S_1|}\frac{(1-p)}{|S_1|} + \sum_{j\in S_2:d_{k,h}} \frac{p}{|S_1|}\frac{p}{|S_2|}\right) \tag{4.26}$$

$$b^k_{S_2^i} = \frac{P^k_{S_2}}{|S_2|}\left(\sum_{j\in S_1:d_{k,h}} \frac{(1-p)}{|S_2|}\frac{(1-p)}{|S_1|} + \sum_{j\in S_2:d_{k,h}} \frac{(1-p)}{|S_2|}\frac{p}{|S_2|}\right) \tag{4.27}$$

where the sums capture the number of elements of each set that satisfy the distance constraint $d_{k,h}$ between the elements of each other set. Since there are only 2 sets in this case we know that one set is observed with probability p means the other is observed with probability $1-p$. We can further define terms to capture the sums more succinctly:

$$|S^{d_{k,h},j}_{l,m}| \equiv \text{number of elements of } S_l \text{ which are distance } d_{k,h} \text{ from element } j \text{ in the set } S_m \tag{4.28}$$

That is, $|S^{d_{k,h},j}_{1,2}|$ translates to the number of elements in set S_2 that are a distance $d_{k,h}$ from the jth element of set S_1. With these terms defined and grouping terms we can then write:

$$b^k_{S_1^i} = \frac{P^k_{S_1}}{|S_1|^2}\left(\frac{|S^{d_{k,h},j}_{1,1}|}{|S_1|}p(1-p) + \frac{|S^{d_{k,h},j}_{1,2}|}{|S_2|}p^2\right) \tag{4.29}$$

$$b^k_{S_2^i} = \frac{P^k_{S_2}}{|S_2|^2}\left(\frac{|S^{d_{k,h},j}_{2,1}|}{|S_1|}(1-p)^2 + \frac{|S^{d_{k,h},j}_{2,2}|}{|S_2|}p(1-p)\right) \tag{4.30}$$

From symmetry, the elements of sets S_1 and S_2 are the same distance regardless of order, so the $|S^{d_{k,h},j}_{1,2}|$ and $|S^{d_{k,h},j}_{2,1}|$ terms will contribute equivalent fractional constraints to the distribution.

In this example, where agent k is located in and observes S_1 and agent h similarly is located in and observes S_2, the second term in $b^k_{S_1^i}$ corresponds to the most likely elements, hence all locations that satisfy the measured distance constraint $d_{k,h}$ are multiplied by p^2. Similarly, the first term in $b^k_{S_2^i}$ corresponds to the least likely elements, where the observations

are exactly opposite of what each agent observes, hence those are multiplied by $(1-p)^2$. The terms where only one observation accurately matches the observed set (S_1 for agent k, or S_2 for agent h) are effectively cross terms here, and are thus each multiplied by $p(1-p)$.

Expressions (4.29) and (4.30) thus capture the essential elements of the multi-agent algorithm proposed. The probabilities from other agents' locations that satisfy the distance constraint are weighted with the agent's own local estimates. This then helps reduce the uncertainty amongst the blocks within a set. However, it must be noted that this assistance is dependent on not only both agents' sensor fidelity p, but also on the fraction of blocks that satisfy the distance constraint out of the total set size: $\frac{|S_{l,m}^{d_{k,h},j}|}{|S_m|}$.

Concretely, if $|S_{l,m}^{d_{k,h},j}| = |S_m|$, then (4.29) and (4.30) will collapse to $\frac{p}{|S_1|}$ and $\frac{(1-p)}{|S_2|}$ multiplied by the prior terms, respectively. Thus all elements of each set would have the same probabilities, or have uniform distributions (the normalization will make sure each remains a distribution). This then reverts to the exact same entropy as the single agent case: $\log|S|$ since $|S_1| = |S_2|$. (To see this more explicitly, simply assemble the elements together, normalize, and plug back into (4.25) and combine terms.) Again, this is unsurprising, since the simple multi-agent protocol of Algorithm 1 only constrains the probabilities by virtue of these distance constraints.

It is then clear that knowing when multiple agents will outperform a single agent requires knowledge of both the sensor fidelity and the particular environment constraints.

4.4.2 Simplified Sensing Environment: Example

Having shown that even in a simplified case, a particular environment is required to calculate the entropy of multiple agents, we now introduce a concrete instantiation of the sensing environment of Section 4.4.1. Figure 4·1 depicts a particular 4x4 environment classified into two sets of equal number elements. Values for $|S_{l,m}|$ (from expression (4.28)) for each

location are shown. We assumed in Section 4.4.1 that agent k is located in S_1 and observes S_1, and agent h is located in S_2 and observes S_2. Let us further specify that agent k is located in element (4,3), and agent h is located in (2,3) of Figure 4·1, thus $d_{k,h} = 2$.

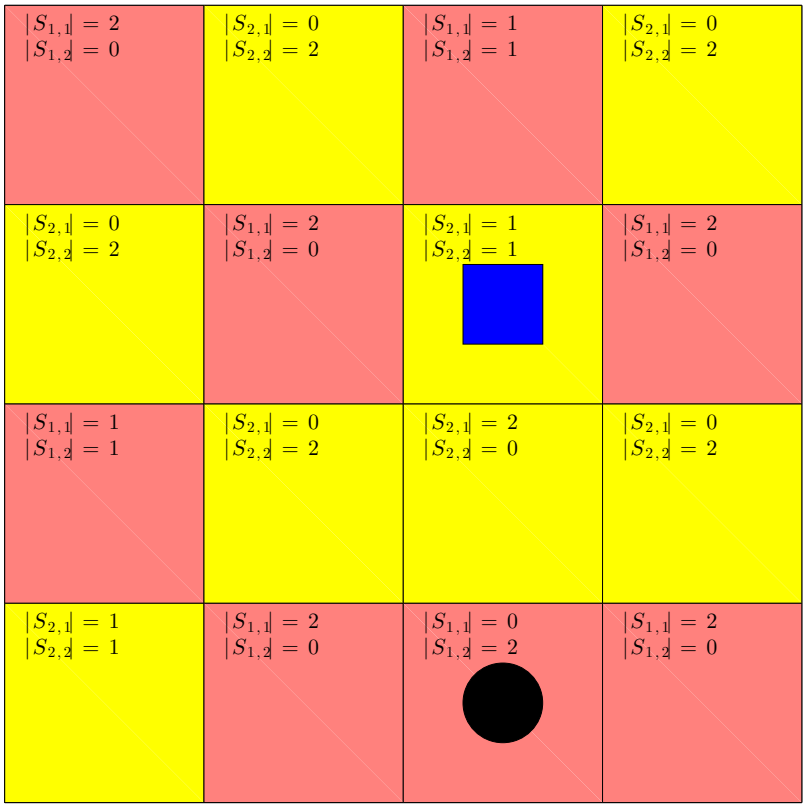


Figure 4·1: Sample environment with number of matching distance elements from each set shown. Note that in this example there are no edge effects, since each tile has exactly 2 neighboring locations which satisfy the distance constraint. Values for $|S_{l,m}|$ (from expression (4.28)) are shown, where red is set S_1 and yellow S_2.

To determine the threshold for where degraded multi-agent sensing would reduce entropy more than a single agent with better sensing we can now calculate out the entropy expressions for both. Combining expressions (4.25), (4.29), (4.30) for multiple agents and (4.23)

for a single agent, we can then write the question of interest as follows:

$$H(b^k_{S_1^1}, ..., b^k_{S_1^{|S_1|}}, b^k_{S_2^1}, ..., b^k_{S_2^{|S_2|}}) - H(b_{S_1^1}, ..., b_{S_1^{|S_1|}}, b_{S_2^1}, ..., b_{S_2^{|S_2|}}) \overset{?}{<} 0 \tag{4.31}$$

where a value less than zero indicates multiple agents have less entropy than a single agent. Expanding, this becomes:

$$\left[h_B(P^k_{S_1}) + P^k_{S_1} H\left(\frac{b^k_{S_1^1}}{P^k_{S_1}}, ..., \frac{b^k_{S_1^{|S_1|}}}{P^k_{S_1}} \right) + P^k_{S_2} H\left(\frac{b^k_{S_2^1}}{P^k_{S_2}}, ..., \frac{b^k_{S_2^{|S_2|}}}{P^k_{S_2}} \right) \right] - [h_B(p)] + \log|S|] \overset{?}{<} 0 \tag{4.32}$$

Grouping terms together gives:

$$\underbrace{[h_B(P^k_{S_1}) - h_B(p)]}_{\text{Distribution of sets}} + \underbrace{\left[P^k_{S_1} H\left(\frac{b^k_{S_1^1}}{P^k_{S_1}}, ..., \frac{b^k_{S_1^{|S_1|}}}{P^k_{S_1}} \right) + P^k_{S_2} H\left(\frac{b^k_{S_2^1}}{P^k_{S_2}}, ..., \frac{b^k_{S_2^{|S_2|}}}{P^k_{S_2}} \right) - \log|S| \right]}_{\text{Distribution within sets}} \overset{?}{<} 0 \tag{4.33}$$

The grouping in expression (4.33) is helpful for seeing how multiple agents could potentially perform better than a single agent with better sensing. In effect, multiple agents would have to reduce the uncertainty within sets more than the perfect sensing of a single agent reduces the uncertainty between the two sets themselves, as we have been describing throughout Section 4.4.1.

Carrying out the calculations in expression (4.33) based on the values for $|S_{l,m}|$, we find that multiple agents with sensing probability of $p \geq 0.93$ will make expression (4.33) true. That is to say, multiple agents with degraded sensing, operating under the assumptions we have described in this particular environment, will have lower entropy than a single agent with perfect sensing.

As noted in Section 4.3, there are circumstances where multiple agents following Algorithm

1 could actually have greater entropy than a single agent when sensing degrades. Figure 4·2 shows how, in this very example, this occurs for probabilities $0.5 < p < 0.79$. However, as shown in the same figure, the MAP estimate difference between the two highest location probabilities $P(\tilde{x}_{(n)}) - P(\tilde{x}_{(n-1)})$ for multiple agents is not only positive for the entire range of probability p, it is strictly increasing in probability p. In this example, the distribution MAP is in the true location for agent k for all probability $p > 0.5$, i.e. it is estimating the location of agent k accurately. Thus, while entropy may not strictly decrease as a function of p in all circumstances for multiple agents, accuracy is still improving. In essence, while the entropy between the distribution of sets in expression (4.33) may favor a single agent more than the reduction within sets from multiple agents, the actual estimation accuracy for multiple agents is still improved.

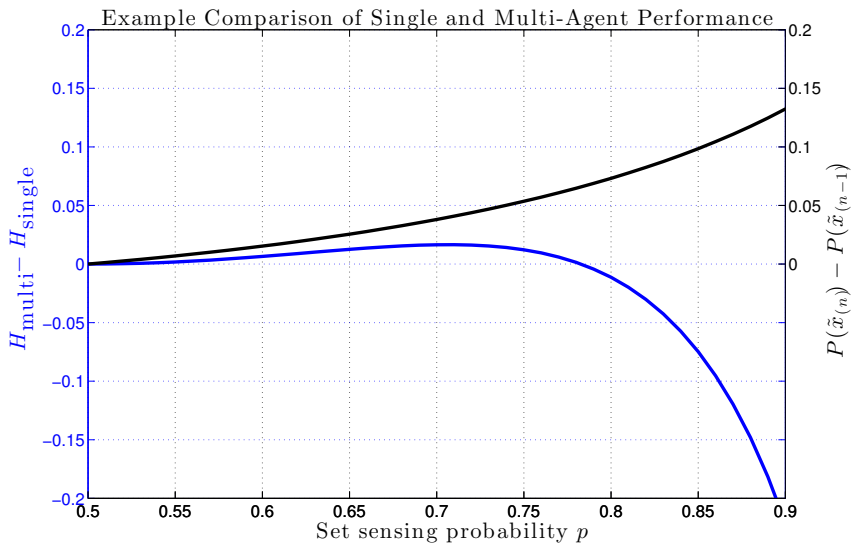


Figure 4·2: Comparison of single and multi-agent performance as a function of set sensing probability p. While the difference in entropy H is not strictly negative, indicating multiple agents can increase entropy, the MAP estimate as compared to the second highest location probability $P(\tilde{x}_{(n)}) - P(\tilde{x}_{(n-1)})$ for multiple agents demonstrates that the actual estimate for multiple agents in this environment remains both accurate and smoothly increasing as probability p increases. This implies that while entropy may not strictly decrease as a function of probability p, the accuracy of multiple agents, in this example, still does strictly increase.

We can also use expression (4.33) to show that, in general, one cannot guarantee the existence of a probability $p < 1$ which will ensure multiple agents do not have greater entropy than a single agent. Consider a situation where, once again VR $= \frac{1}{2}$, where now the vision sensor gives probability $p = 1 - \epsilon$ of observing S_1, and probability ϵ of seeing S_2. Now consider that in this particular environment, the measurement constraint from another agent uniformly shifts a small amount of probability mass totaling δ to each element of set S_2, leaving every element within both sets S_1 and S_2 equiprobable within their respective sets. From expression (4.33) we would write:

$$\underbrace{h_B(1-\epsilon-\delta) - h_B(1-\epsilon)}_{>0} + \underbrace{(1-\epsilon-\delta)\log|S| + (\epsilon+\delta)\log|S| - \log|S|}_{=0} > 0 \qquad (4.34)$$

which immediately shows that multiple agents will result in an increase in entropy from a single agent. To be sure, this situation may be quite uncommon, if not practically impossible with some reasonable additional constraints on the environment, and does not necessarily correspond to degraded localization performance.

Concrete examples of challenging VR=$\frac{1}{2}$ environments where multiple agents are guaranteed to not increase entropy are shown in Figure 4·3 and the corresponding reduction in entropy is shown in 4·4, demonstrating that one need not be overly concerned by this negative result. However, it does reflect the inherently challenging nature of profoundly perceptual aliased environments where only limited sensing is available.

4.5 Discussion

The primary purpose of this Chapter has been to breathe some mathematical intuition into the algorithm presented in Chapter 3 by doing some first-step analysis of single and multiple agents localizing within in a categorized map. We have considered simple cases to demonstrate the ideas in a more straightforward manner. Indeed, once agents are moving, priors will shift and distributions become more diffuse. Moreover, the need for odometry to help constrain distributions for both single and multiple agents in this type of environment

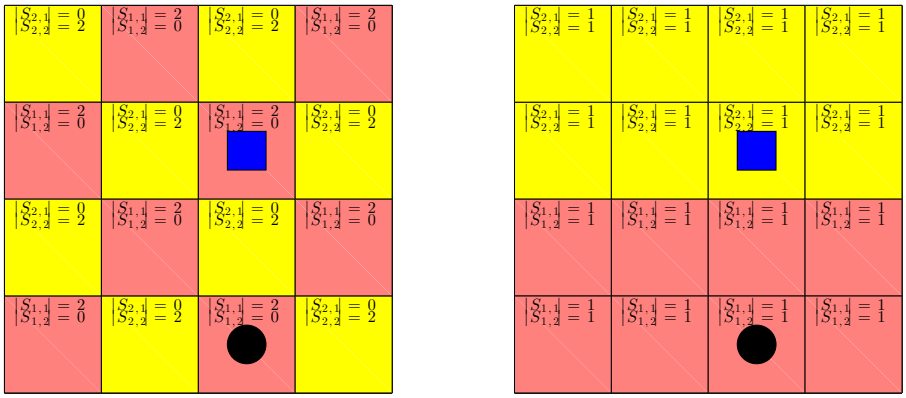


Figure 4·3: Example of challenging VR=$\frac{1}{2}$ environments where multiple agents are guaranteed to not increase entropy. In the checkerboard case, multiple agents in fact reduce entropy generally for probabilities $p < 1$, as shown in Figure 4·4.

is quite clear.

The concept of entropy is a useful tool, even if it may not always explain the entire story, as demonstrated here. A potential topic of further investigation would be using the entropy of a semantic classifier to approximate the visual richness (VR) of an environment. The entropy of the output layer of a semantic descriptor, such as Places-CNN, has a similar interpretation to the VR of an image. The more spread-out and uniform the classification distribution, the more semantic uncertainty and entropy in that image, and environment more generally.

Having demonstrated the principle of a basic visual multi-agent localization algorithm in example perceptual aliased environments, Chapter 5 will explore how this model can be brought to bear in a more realistic setting, using satellite and in-flight camera imagery, and incorporating nonholonomic motion and odometry constraints.

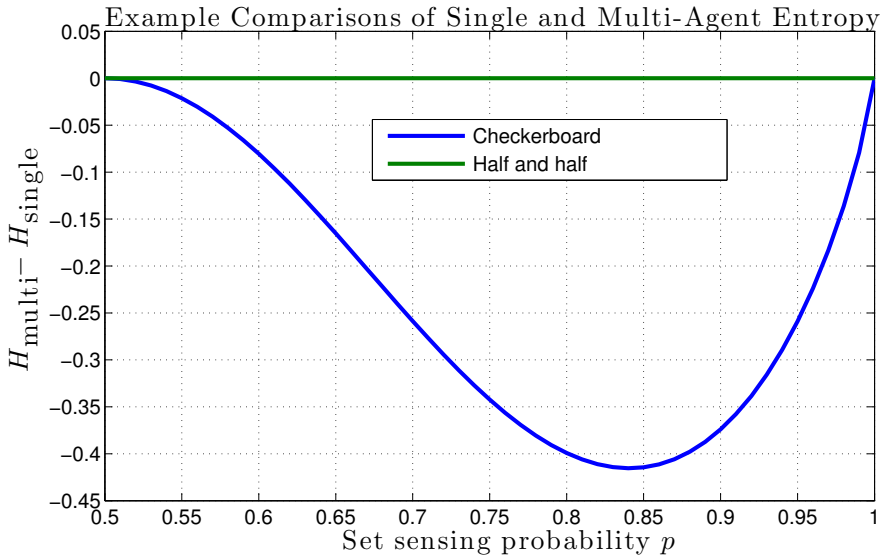


Figure 4·4: Comparison of single and multi-agent entropy for the examples shown in Figure 4·3. In the checkerboard case multiple agents help reinforce the certainty of being in S_1 (red) for probabilities $p < 1$, whereas in the half and half case they are not able to provide any additional reduction since all blocks satisfy the measured distance constraint. In both cases, however, entropy for multiple agents does not increase as compared to a single agent, demonstrating that Algorithm 1 should not be seen as generally detrimental for degraded sensing cases.

Chapter 5

In-flight Camera Imagery Testing

Having introduced the basic notion of multi-agent localization using a discretized environment to enable semantic image recognition and representation, a concrete demonstration of this methodology is now presented with actual terrain images from satellite and in-flight camera imagery from Eloy, AZ.

A number of modifications from sections 3.4-3.6.2 are now detailed.

5.1 Semantic Classification Using Convolutional Neural Networks

In Section 3.4.1 and the following example implementation in sections 3.4.2-3.6.2, the notion of semantically classifying imagery into sets was introduced. In general, however, real-world imagery often falls into multiple semantic categories. A natural way to capture this idea in a probabilistic framework is to use a convolutional neural network (CNN) trained on imagery of "places". As described in Section 3.3, the Places-CNN network [62] does just this, classifying input imagery into a distribution of 205 categories of places.

The following sections describe the test data, along with how a probability distribution $P(x|z)$ is generated for each location x given image measurement z.

5.2 Satellite Map Data

The satellite map represents a 10.28 x 10.28 km area, which is divided into a grid of 10 x 10 tiles (Figure 5·1). The satellite tiles are processed through the Places-CNN classifier using Caffe (Section 3.3), and the distributions for each tile are stored in a database (100 tiles

x 205 category probabilities). This 100x205 array represents the entire pre-computed map stored on-board that in-flight imagery is compared against.

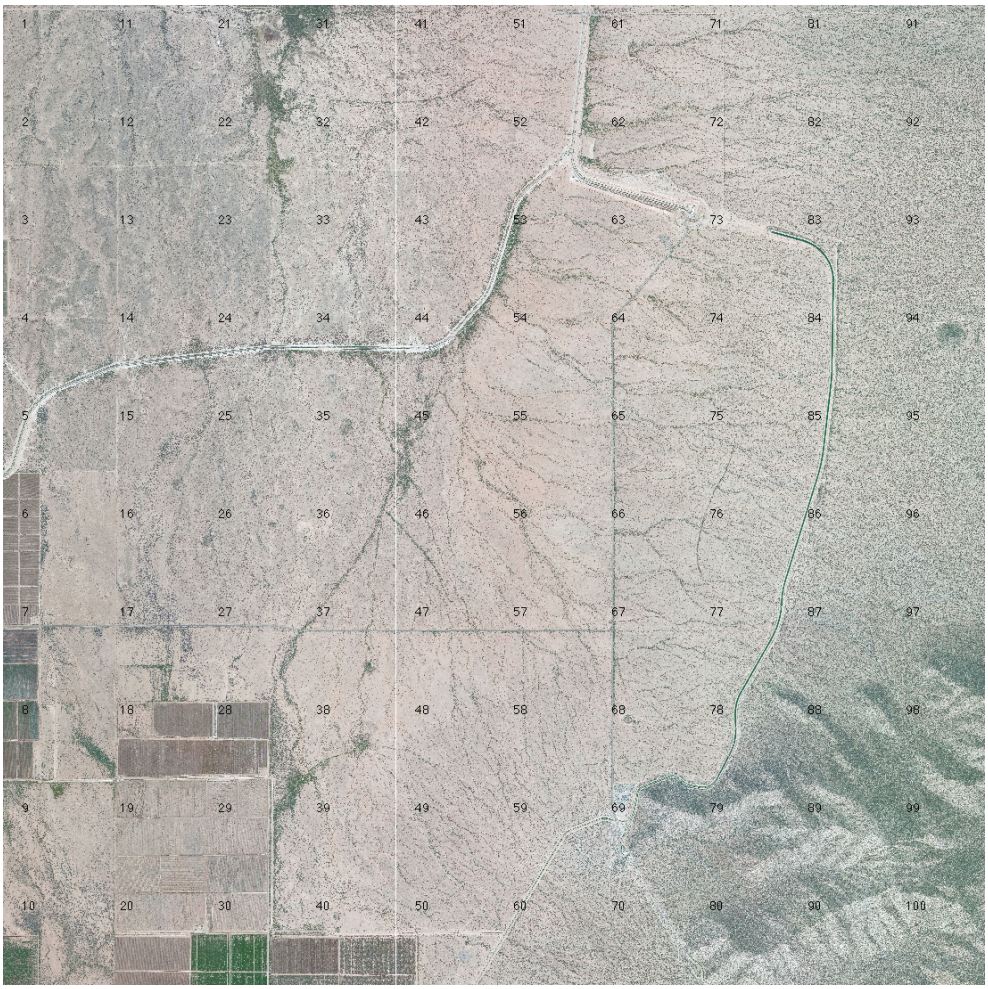


Figure 5·1: Satellite image of 10.28 x 10.28 km area in Eloy, AZ divided into a 10x10 grid. Note the presence of roads, mountains (bottom right corner), fields (bottom left corner and left side) and large amount of desert.

In practice, it should be noted that the Places-CNN network contains many more categories than would be visible in a terrain navigation context (indoor, underwater, or other places). Figure 5·2 shows the average distribution values for the satellite imagery in Figure 5·1 among the top 50 categories, demonstrating this quite clearly. Thus the representation used here should not be seen as minimal. Rather, it represents the ability to take an off-

the-shelf CNN and use it for a purpose it was not expressly designed for–a new network trained specifically with satellite imagery could almost certainly produce better results.

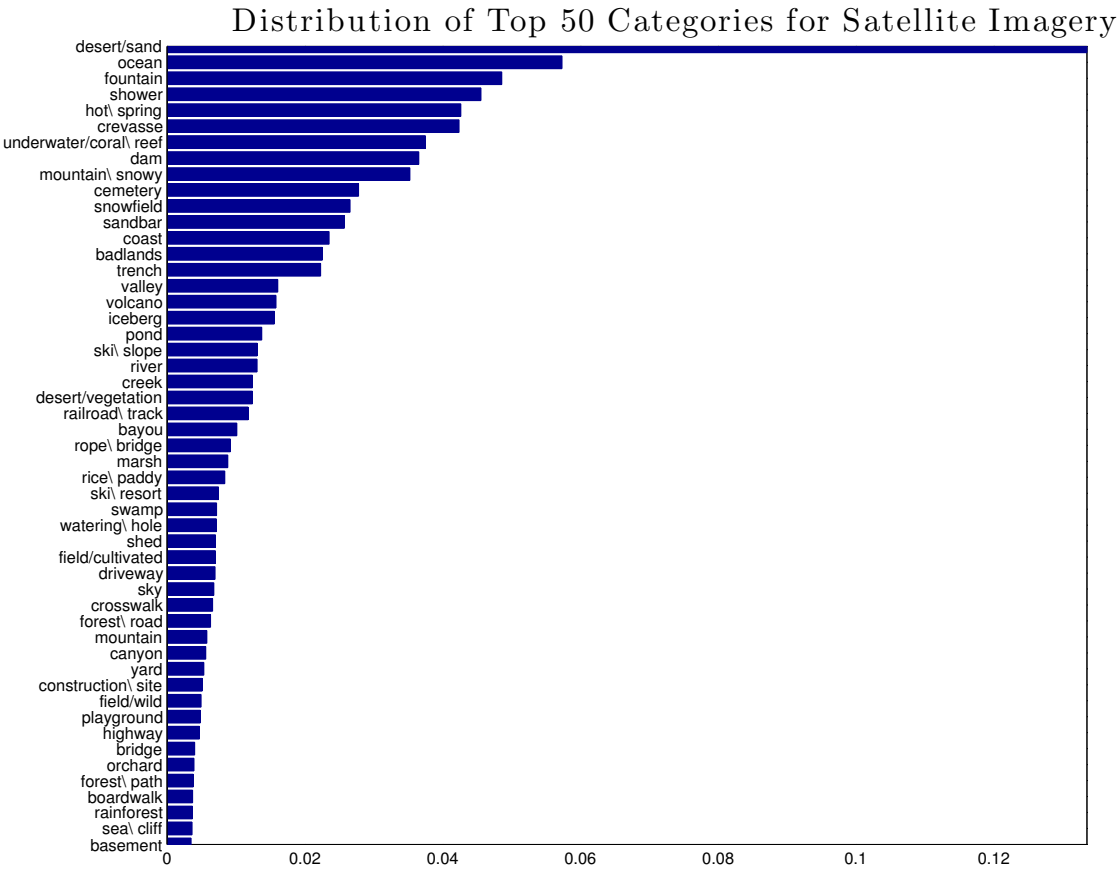


Figure 5·2: Distribution of top 50 Places-CNN categories for satellite imagery in 5·1. Even of the ostensibly relevant outdoor categories, very few have significant probabilities in this data, indicative of a non-minimal representation that could be improved upon with more targeted neural networks.

5.3 In-flight Camera Imagery

In-flight camera imagery from multiple airdrop flight tests is projected to the ground frame and similarly divided into corresponding grid tiles (Figure 5·3). Figure 5·4 shows a visual-

ization of a complete map composed entirely of camera images projected this way. Some map locations (particularly the corners and very center of the map) are not entirely visible from the available in-flight imagery, thus the camera images of those tiles capture only subsections of the corresponding ground area. These images are similarly classified using the Places-CNN descriptor, for comparison with satellite imagery.

As detailed in Section 5.5, discrete heading states are introduced to each location state to better capture the nonholonomic dynamics of the parafoil system. As such, the camera imagery the vehicle observes is heading dependent. For testing purposes, a camera image corresponding to a particular grid tile location is returned from a database of multiple camera images from different flights, rotated to correspond to current agent heading (i.e. images from the same location will vary based on heading). Practically, this means that as agents navigate the simulated environment they may observe any of a number of heading dependent camera images at each location–a realistic scenario of unknown location and heading.

It should be noted that the camera tiles in Figure 5·4 are representative of how a real agent might observe the environment, with some landmarks being classified in different tiles than the satellite map of Figure 5·1–in essence, the tile designation should not be seen as overly limiting. The assumptions in Section 3.1 of altitude and sufficient IMU estimated knowledge to generate a nadir (downward), image orientation scaled appropriately are used to generate camera imagery identical to the data used in simulation here.

5.4 Generating distribution from images

To generate a probability distribution from an input camera tile image z to a satellite tile image location x, an approximate nearest neighbor (ANN) algorithm [3] is used to generate distances d_x between the Places-CNN classifiers for the input image and each satellite image

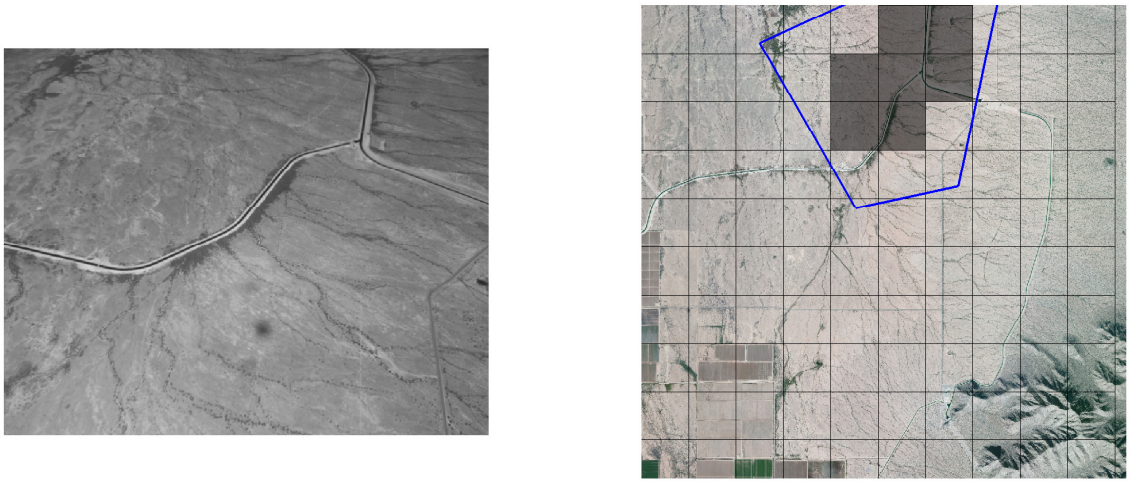


Figure 5·3: In-flight camera image (left) projected to ground plane tiles (right). This procedure is how camera data is generated for testing against satellite classification.

tile x. This is converted to a distribution quite simply:

$$\tilde{d}_x = d_x + \delta \tag{5.1}$$

$$P(x|z) = \frac{1}{\tilde{d}_x * \Sigma_x \frac{1}{\tilde{d}_x}} \tag{5.2}$$

where δ is a small term to ensure $\tilde{d}_x > 0$.

This procedure thus supersedes the update expression (3.6) for generating a distribution $P(x|z)$.

5.5 Nonholonomic Motion Model

In Section 3.6.1 the agents described are free to move in any direction to any of the adjacent tiles at each time step. Parafoil systems are constrained by nonholonomic unicycle-like motion, however, and as such heading states are introduced. In keeping with the simple discretization approach, each location is composed of 8 heading states (NE, E, SE, S, SW,

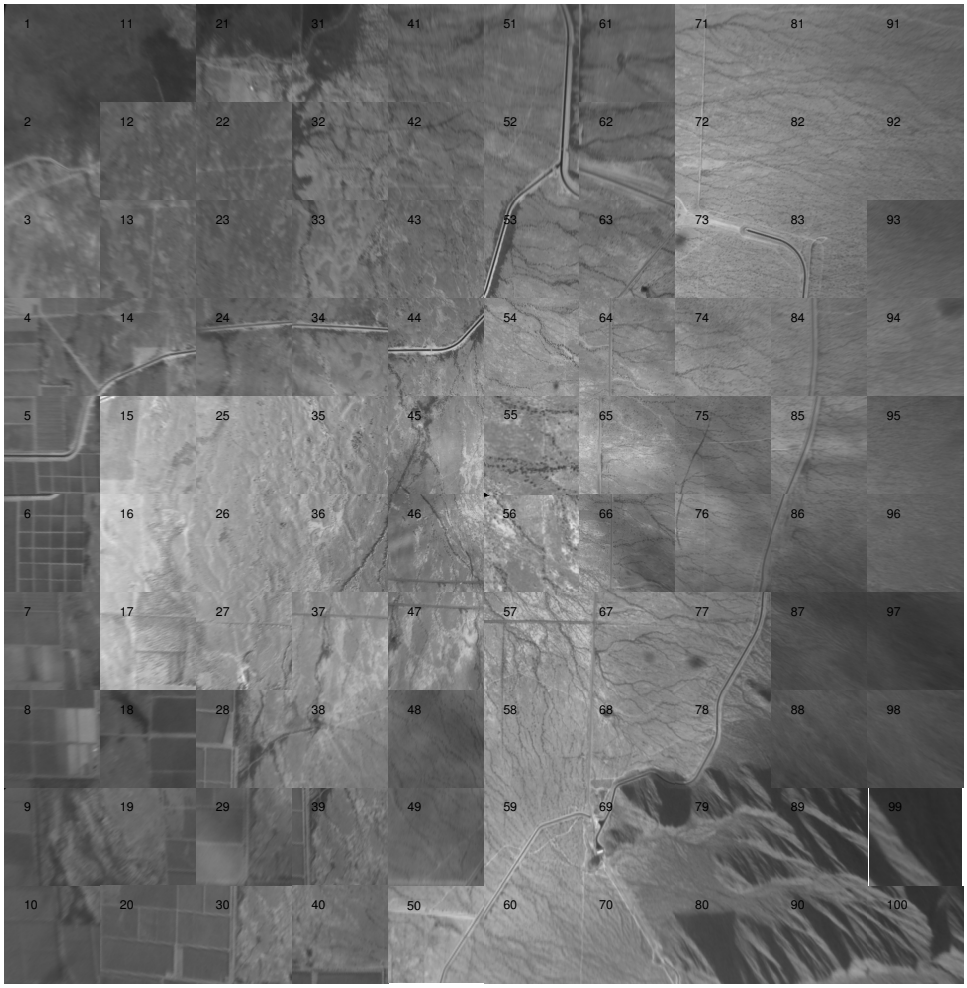


Figure 5·4: Map of similar area as shown in Figure 5·1, with tiles composed entirely of camera images projected to the ground plane (Figure 5·3).

W, NW, N). Vehicle motion at each time step is then constrained to, at most, a single rotation and translation. This is depicted in Figure 5·5. The model remains Markov, thus expression (3.11) remains accurate, repeated here:

$$x_k(t+1) = Tx_k(t) \tag{5.3}$$

where T is a left stochastic matrix. This defines the probability distribution $P(x_k(t+1)|x_k(t))$ for each agent k in expression (3.7) in Section 3.4.2.

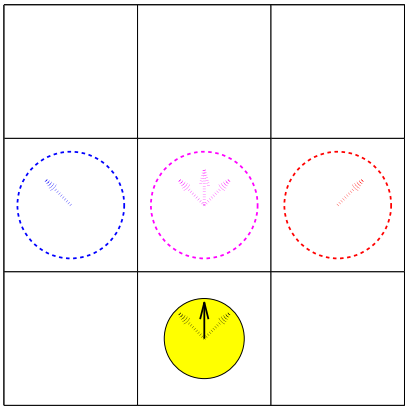


Figure 5·5: Depiction of legal moves from a given state. There are at most 8 moves possible from each state, consistent with unicycle-like nonholonomic motion constraints.

This notion of multiple heading states at each geographic location is not only useful for capturing more realistic vehicle motion, but also allows for a natural implementation of odometry (Section 5.6) and heading-dependent camera imagery (Section 5.3).

5.6 Odometry

Each agent is assumed to have probabilistically accurate local body-frame odometry from one step to the next to be aware of which of the (at most 8) legal moves it made in the previous time step for calculation of the predict equation (3.7). Concretely, this means if the vehicle made a turn to the right followed by a translation forward, the agent is aware it did this relative to its previous pose, but not aware of how this motion is reflected in a global frame (i.e. no explicit global heading information). This is implemented by changing the weights in matrix T in expression (5.3) from being uniformly distributed between all possible legal moves, to instead favoring the states consistent with the vehicle body-frame motion. This is consistent with the advances in odometry accuracy described in Section 2.2.

While the vehicle does not have explicit absolute heading sensing, over time this odometric information implicitly enables the agent to estimate both location and heading as it travels within the environment. In essence, this combination of discrete heading and location states represents a way of tracking heading and location with a single state estimator.

It should be pointed out that the existence of edges and corners in this environment should not be seen as limiting. The available odometry does not immediately allow estimates to converge when an edge or corner is reached–rather the agent is only aware of what motion it made, not what possible ones were available. Moreover, as demonstrated experimentally in Section 3.6.2, an edge-free scenario produces similar results to a environment with edges. Furthermore, the mathematical development in Section 4.3 confirmed that edges are not required for multiple agents to constrain their distribution more than a single agent. In practice, the boundaries serve a similar purpose to a basic guidance algorithm that would ensure an agent does not travel too far away from other agents or good landing sites it has observed (Chapter 6).

5.7 Multi-agent Cooperation Protocol

As described in Section 2.4, maintaining statistical consistency is one of the major challenges in extending localization algorithms correctly to a multi-agent context. In Section 3.6.1, the notion of agents sharing sets was used to decouple the distributions communicated between agents to a large extent, making successive measurements very weakly dependent.

For the experiments considered here, however, completely independent measurements are used. This is achieved quite simply: each agent k simply broadcasts the distribution $P(x_k(t)|z_k(t))$ of its current location x_k based on its latest observed image z_k at current time step t, as described in Section 5.4, with no additional processing. That is, the output of the CNN distribution of the image z_k is compared to that of the satellite database to generate a distribution, and that information is broadcast, without being processed through

the predict step (i.e., not combined with the running estimate based on previous image measurements or odometry). This is probabilistically equivalent to each agent broadcasting the raw image it observes, or the output of the Places-CNN classifier itself.

The choice of broadcasting the distribution created from the raw information is merely to save extraneous computation. In a scenario where a common description of observed imagery is desired, the semantic Places-CNN output distribution could easily be broadcast instead, and would remain a consistent message size regardless of number of possible locations. Indeed, in a scenario where the number of locations is greater than the Places-CNN output (i.e. more than 205), transmitting the CNN output itself is preferable, particularly since this semantic information could be used by other agents (including human operators) more universally. There is no handshake or message received step, since no book-keeping is required of which agents received which messages from which other agents.

While in the real world it is possible that some agent's cameras will operate differently, generating slightly dependent distributions from one time step to the next, this approach represents the closest to truly independent measurements possible in this framework. In practice, this translates to transmitting a vector of values with cardinality of total locations (in this case a 100 element vector) which correspond to a distribution (the elements sum to 1), or, equivalently, a vector of the Places-CNN output distribution (205 elements).

As in Section 3.6.1, these measurements are used by each agent k by combining scalar distance information $d(x_k, x_h)\ \forall\ h \in K$, available by measuring signal strength of messages received, for instance (RF-ranging), with these distance measurements (assumed to have bounded noise ε). Given the assumptions of Section 3.1 of estimated altitude knowledge, agents at different altitudes can still calculate horizontal plane distance from these measurements with a simple Pythagorean calculation. This enables agents to reweight the probabilities of their own current estimate using the broadcast information from each other

agent's distribution geometrically, as described in Section 3.6.1.

It is worth underscoring the extremely limited form of multi-agent communication and coordination this represents. Agents broadcast a compressed form of their own current, independent, observations (equivalent to raw images) into the ether for any other agents within range to use, based solely on roughly measuring the scalar distance away the broadcast originated from. The maximum amount of transmitted data from each agent is the output layer of the semantic classifier, being a 205 element vector of in this case.

Moreover, this communication protocol, modified from Algorithm 1, is robust to intermittent or dropping messages: agents will simply proceed with the available information, without concern for estimating any other agent's state, or keeping track of which messages originate from whom. The algorithm will accommodate failures from other agents gracefully, without rebuilding an entire filter or factor graph when messages from some agents suddenly go missing, and just as suddenly reappear. This is a crucial capability for parafoil systems, where an unknown number of agents will be deployed at once in potentially chaotic environments, with malfunctions and interference common; other algorithms (such as those described in Section 2.4) that require updates from all agents at each time step will not be suitable.

5.8 Adjacent Tile Sensing

As depicted in Figure 5·3, at higher altitudes the camera image often contains multiple tiles. While varying tile sizes as a function of altitude could be used to capture this, the resolution of the localization is inherently limited by the choice of tile size (as one would expect). Thus, the notion of sensing adjacent tiles is introduced to capture the concept of seeing multiple tiles at once. Figure 5·6 depicts 4 adjacent sensing depths. Zeroth order represents seeing only the tile the agent is immediately over, reverting to the case of no adjacent tile information. First order adds the (at most) three additional tiles that the

agent may navigate to next (following from its current heading), and depths two and three extend those locations outward in the direction of current heading. As in the case for multi-agent information, geometric constraints are used to incorporate measurements from adjacent tiles. To mimic real-world conditions, as simulation time proceeds the adjacent tile depth sensing decreases as the remaining flight time (and altitude) decreases. Each depth level runs for one quarter of simulated flight time steps, starting from depth three and decreasing to until depth zero. A more precise model based on sink rate and current altitude could be developed, but this represents a simple method for capturing how the additional information from adjacent locations can help agents localize.

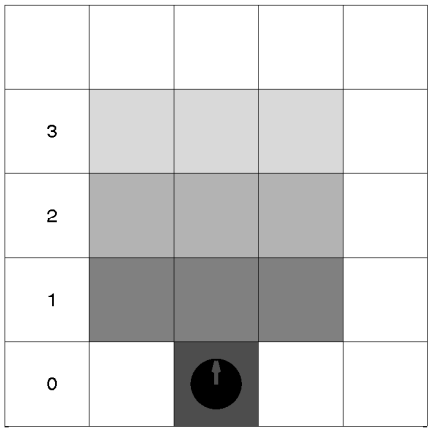


Figure 5·6: Depiction of adjacent tile sensing, depths 0 (current tile) through 3 (9 additional tiles). This captures the effects of varying altitude during flight, as the levels of adjacent sensing available recede in consonance with flight time to model agents descending.

5.9 Simulation Setup and Results

To test the setup as described in Sections 5.1-5.8, 1000 simulations of 400 time steps are run. As in Section 3.6.1, the agents move through the environment by taking a random walk, choosing one of the (at most) 8 legal moves (Figure 5·5) at each time step arbitrarily. The results are shown in Table 5.9 and Figures 5·7, 5·8, 5·9, 5·10, and 5·11. Figure 5·7 shows

	Number of Agents			
	1	2	3	4
% Correct (⇑ better)	3.615	18.28	29.34	32.01
% Distance Error (⇓)	37.44	35.62	33.04	32.67
% Heading Error (⇓)	29.0	27.31	25.23	24.99
% Steps to First Match (⇓)	82.15	30.48	12.64	6.90
% Entropy remaining (⇓)	55.55	12.21	1.88	1.02
MAP confidence (⇑)	0.1102	0.613	0.7837	0.8298

Table 5.1: Localization summary results using Places-CNN classifier in 10x10 discretized environment.

	Number of Agents			
	1	2	3	4
% Correct (⇑ better)	1.29	5.82	15.32	21.97
% Distance Error (⇓)	36.19	37.49	35.10	33.70
% Heading Error (⇓)	30.79	31.95	29.62	28.51
% Steps to First Match (⇓)	95.77	74.66	30.64	17.06
% Entropy remaining (⇓)	41.0	17.36	6.65	2.28
MAP confidence (⇑)	0.0248	0.2394	0.6225	0.7745

Table 5.2: Localization summary results using Places-CNN classifier in 20x20 discretized environment.

the fraction of correctly estimated locations as a function of number of agents. Figure 5·8 shows the number of steps it took in each scenario to generate the first accurate match. Figures 5·9 and 5·10 show the average normalized location and heading errors, respectively. Figure 5·11 shows the relative certainty of estimates at each time step for correct estimates, and Figure 5·12 shows the fraction of remaining entropy for each scenario. These results are discussed in Section 5.10

For comparison, a simulation is set up with the satellite and camera space discretized into 20x20 (400 tiles) to demonstrate the effects of scaling. Everything carries over from the description of the 10x10 scenario (agents can still only move to adjacent tiles, observe adjacent tiles, etc). Results for 100 runs of 400 times steps are summarized in Table 5.9.

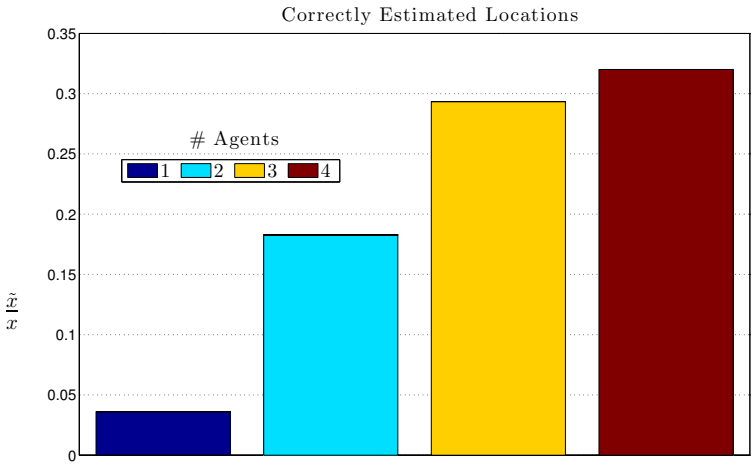


Figure 5·7: Fraction of accurately estimated locations $\left(\frac{\hat{x}}{x}\right)$. Increasing agents significantly improves estimation accuracy. This Figure (and Figures 5·8 through 5·12) are generated for a 10x10 environment with 8 headings each (800 possible states) classified using the Places-CNN descriptor of camera imagery compared to satellite imagery. For each sensor and number of agents combination, averages are shown for 1000 simulations of 400 time steps each.

5.10 Discussion

In general, the results follow the pattern demonstrated in the simple example in Section 3.6.2, with an overall very clear theme: multiple agents make a significant difference. Figure 5·7 is remarkably similar to Figure 3·3 for the case where sensor accuracy for identifying the correct image set is 60% (see the first column of Figure 3·3). This is consistent with the noisiness observed in matching the satellite and camera imagery generally, where the camera image for a particular location is often classified quite differently from a satellite image of the same location. It is with the combination of odometry and multi-agent information that the distribution can be constrained sufficiently to provide more useful results in accurately estimating vehicle location and heading. Figure 5·11 demonstrates the relative certainty of correct guesses that the agents have in each scenario, again highlighting the great advantage of multiple agents. Figure 5·12 depicts the amount of remaining entropy in each agent's distribution as a fraction of starting entropy. Consistent with the develop-

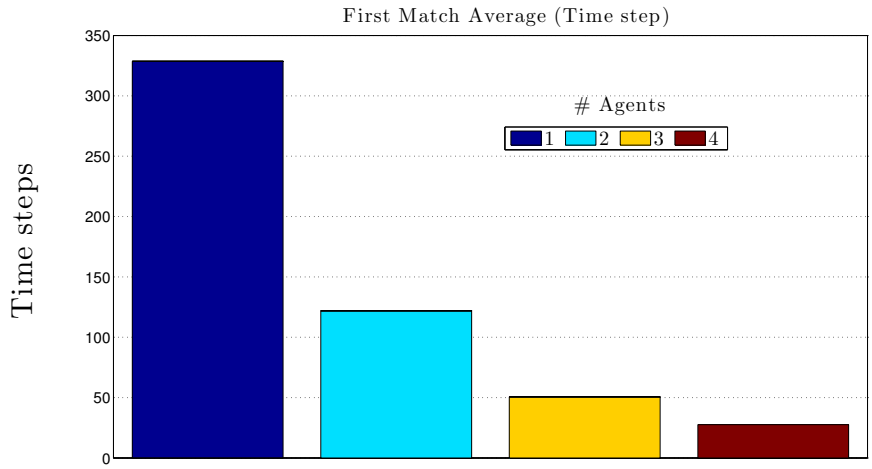


Figure 5·8: Number of time steps to achieve first match (location and heading). The length of time required for the first accurate match with a single agent is striking, but unsurprising, given the noisy camera data and no explicit heading estimation.

ment in Chapter 4, multiple agents will constrain the probability distribution of location far better than a single agent–not simply estimating location correctly, but doing so with much less uncertainty in the entire distribution.

Of particular note is Figure 5·8, where the single agent case takes a large portion of the entire flight time to find a match. The implication of this is that adjacent sensing (Section 5.8), while theoretically providing more information, can be confounding when no additional information is available to help constrain the distribution.

It is important to point out, however, that in Figure 5·9 and 5·10, the actual errors for incorrect matches in location and heading are not as dramatic as one might expect from Figures 5·7 and 5·8. The implication is that even in the single agent case, while the precise location and heading are not accurate, the algorithm is still able to weight the distribution in the general vicinity of the actual location, thus even a single agent using this approach would not be completely lost, but rather have a better general sense of where it is located

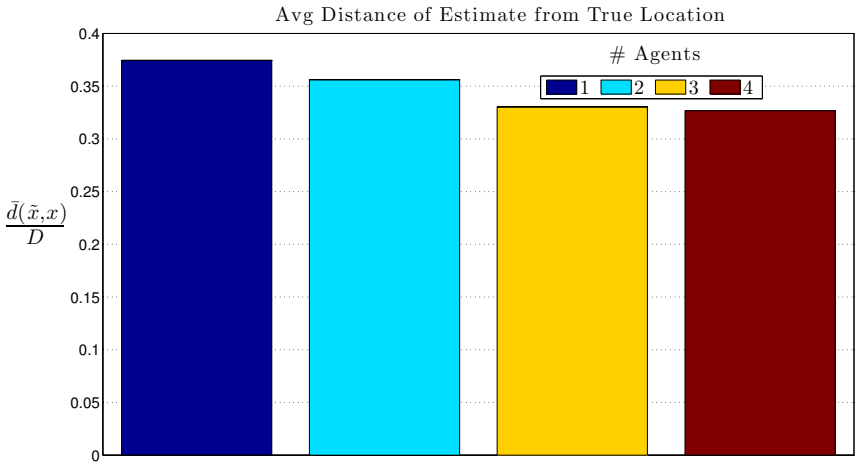


Figure 5·9: Normalized average distance $\left(\frac{\bar{d}(\hat{x},x)}{D=\max(d_{ij})}\right)$ of estimate from actual position as a function of adjacent tile sensor depth (lower is better). This shows how far off, on average, the location estimate is for the scenarios in Figure 5·7. The performance of a single agent is more reassuring here, demonstrating that while getting the precise pose may be much more challenging, awareness of the general vicinity is more feasible for even a single agent.

in an environment, which can be extremely useful as well.

For these simulations, the agents are following a random walk, whereas in practice, agents would likely be following a more logical search strategy. The random walk guidance strategy is useful, however, in demonstrating the power of the localization algorithms to function as designed, even with a sub-optimal (random) search approach.

Turning to the 20x20 simulation results, performance in all aspects degrades, as expected. In this environment, even two agents have great difficulty accurately identifying locations, and, in fact, have slightly degraded performance in terms of distance and heading error as compared to a single agent. Two agents operating in this environment do clearly constrain their distribution entropies, find a first match quicker, and have greater accuracy, but is not until the three agent case where dramatic performance improvements across the board are observed, rather than two agent case in the 10x10 case. Indeed, the performance in the

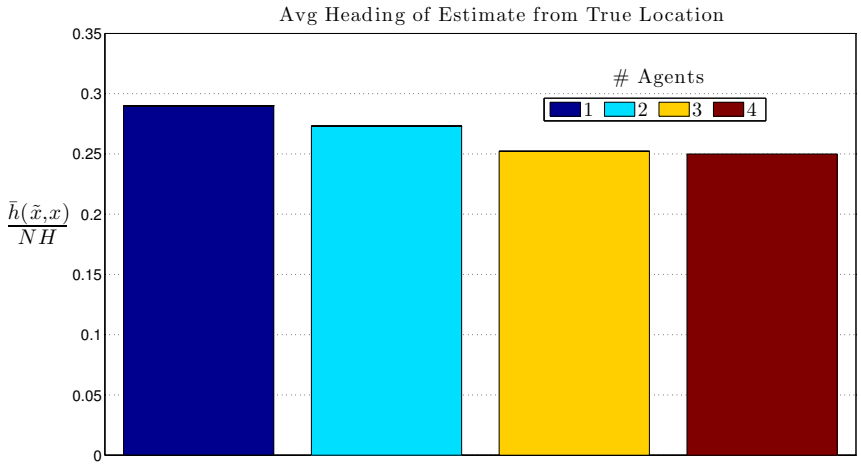


Figure 5·10: Normalized average heading error $\left(\frac{\bar{h}(\hat{x},x)}{NH=\#\text{ heading states}}\right)$ of estimate from actual position as a function of adjacent tile sensor depth (lower is better). This shows how far off, on average, the heading estimate is in each of the scenarios described in Figure 5·7.

case of three agents in the 20x20 environment is quite similar to the performance in the case of two agents in the 10x10 environment. All of this is consistent with the development in Chapter 4, where the ability to constrain the environment successfully is dependent on environmental constraints as much as sensor fidelity.

As far as the limited accuracy with in-flight camera imagery, it should be noted that the limiting factor is not the approach introduced here, but rather the underlying ability of current ML techniques to accurately classify imagery from different cameras and viewpoints consistently, along with the challenging VR of the environment itself. As ML technology continues to progress, whether with new CNNs or other approaches, the sensing data available should improve, as will performance of the algorithms presented here (as demonstrated in Figure 3·3).

Regarding real-time feasibility, calculation of the Places-CNN classifier for an input image, along with calculation of the approximate nearest neighbor distribution combined is less

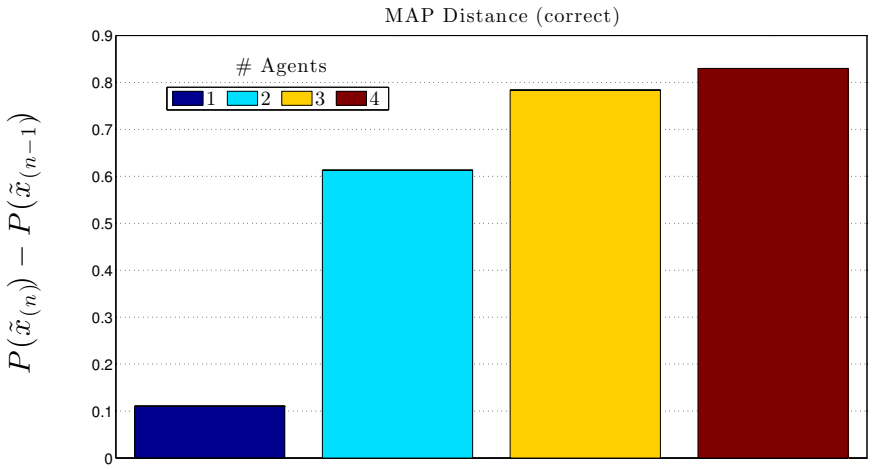


Figure 5·11: Average distance between probability of MAP and second highest element of the distribution. This demonstrates the relative certainty of the MAP estimate in each scenario. Once again, multiple agents dramatically improves this metric, meaning not only are locations determined more accurately, but there is more certainty in those determinations.

than a 300 millisecond operation on a consumer grade laptop (Intel Core i7 with Nvidia 750M GPU).

The approach presented here has a type of scale and time invariance. A map of any size can be divided this way, into arbitrarily large regions, determined by the desired application and localization requirements (planetary landing, low earth orbit, navigation assuming flat-earth, etc). The tiles need not be of uniform dimension, and can be sized based on any number of criteria, as demonstrated in the Google PlaNET [60] representation. The only requirement is the definition of metric distance between tiles.

Time steps are also subject to scaling, and need not be at any particular rate for this framework to be useful (even if the processing operations can be done very rapidly). In the case of guided parafoils with flights lasting multiple minutes, a reasonable time step could be 5 or even 10 seconds, and for other applications potentially longer. The time step can be thought of as a reasonable time scale for the particular vehicle dynamics and camera

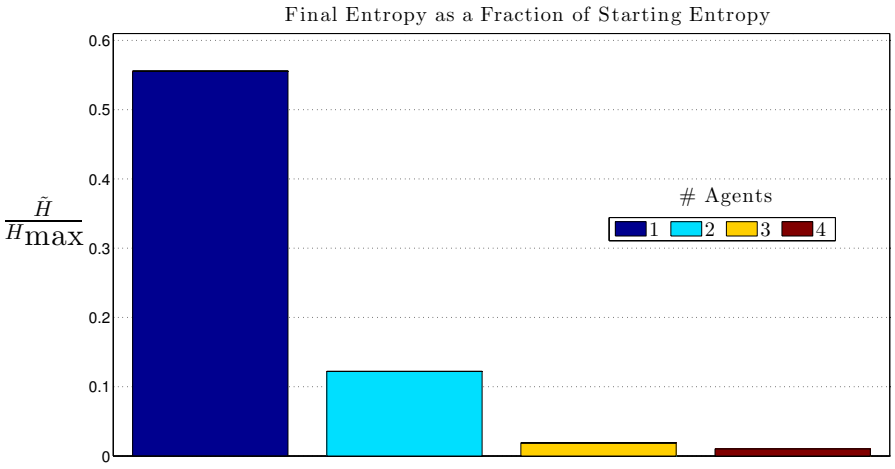


Figure 5·12: Average remaining entropy $\tilde{H}$ as a fraction of starting entropy $H_{\max}$. Multiple agents drastically improves this metric, meaning the entire final probability distribution of location is far more constrainted. This dovetails with the mathematical foundation described in Chapter 4.

processing. The approach presented here is thus fairly general and generic, and can be scaled up or down as localization demands and processing capabilities vary for different applications.

The results shown here demonstrate the great advantage of multiple agents for localization, and do so in a way that is realistic for guided parafoil systems. These systems have limited computational power and on-board storage, along with finite communication bandwidth, a high likelihood of dropped or lost packets, and commonly suffer potentially catastrophic hardware failure. An algorithm that requires all agents to communicate large amounts of information constantly, and requires a complicated book-keeping scheme, will not be able to recover easily from intermittently malfunctioning or failing agents somewhere in the group. Fielded parafoil system commonly experience hardware failures, whether from overworked servo motors, system shutdowns from heat, or sudden refusals to communicate; an approach that does not fail gracefully when these inevitabilities occur will be of limited use. One of the strengths of the approach presented here is the robustness to all manner of these fail-

ures, by intentional choice of a minimalist approach.

What becomes abundantly clear is that in numerous environments, visual localization remains a major challenge. For parafoil systems in particular, this challenge is acute, since solving the localization problem is, by definition, the only way to land in globally specified coordinates. If localization fails, or occurs too late in flight for agents to land at a specified location, there is currently no failsafe in place. This idea has motivated the use of semantic classification in this Chapter, since that very same tool can be used to address this important landing site limitation: the topic of Chapter 6.

Chapter 6

Landing Site Identification

State of the art guided parafoil landing relies on pre-determining and pre-programming ground targets. Each step of this process imposes limitations on the scenarios in which these systems can be deployed. Maps may be unreliable or unavailable. In conflict zones or natural disaster areas, landscape can change rapidly and dramatically. Requiring accurate maps for determining landing sites a priori may delay critical supplies in time-sensitive conditions. Even with accurate maps, fast-changing landing sites may be programmed incorrectly, which commonly happens in real-world flight testing.

Fundamentally, if landing locations are specified in global coordinates, this demands each parafoil successfully localize every flight for success. Particularly in a GPS-denied situation, this may be an onerous requirement for an unpowered parafoil with finite flight time. There is no fail-safe in place should the localization process fail.

Additionally, when there are undesirable landing areas, whether mountains, bodies of water, or other regions that would compromise the survivability and accessibility of payloads; they must both be accurately mapped in advance and stored on-board as keep-out zones, which on-board guidance algorithms work to avoid. This also demands global localization in order to make use of these geotagged landing hazards, and has no adaptability to potentially volatile situations in the real world.

These challenges motivate the idea of using vision to not merely replicate GPS localization, but to enable understanding and classification of the landing environment. Visual navi-

gation should be better, not just more complicated than GPS. The idea, as introduced in Section 3.2, is to use a semantically meaningful visual classifier. Convolutional neural networks (CNN), discussed more in depth in Section 3.3, represent a state of the art machine learning (ML) technique to accomplish this task of visual understanding. In particular, the Places-CNN [62] is designed to identify different types of outdoor places, and has been demonstrated to be useful for classifying satellite and camera imagery for localization purposes in Chapter 5.

6.1 Landing Site Classification Using Places-CNN

The Places-CNN network takes an input image z and outputs a probability distribution d_z classifying the image into n possible different categories of places ($n = 205$ in this case). Formally, $\{d_z \in \mathbb{R}^n : d_{z_i} \in [0,1]\ \forall i \in n,\ \Sigma d_{z_i} = 1\}$. In a parafoil landing context, some places are more desirable landing locations than others. In particular, roadways are often the target landing sites to ensure accessibility and survivability of the payloads. Bodies of water, mountains, and other treacherous areas are to be avoided, for the same reasons.

A method for incorporating both desirable and hazardous landing sites is readily apparent when semantic classification of imagery is available on board: weight preferred landing sites favorably, and undesirable landing sites unfavorably in a vector $w \in \mathbb{R}^n$. Then, for each image processed through the Places-CNN, a classification goodness score $g_y \in \mathbb{R}$ for landing suitability of location y can be immediately obtained by a simple dot product:

$$g_y = (d_z)^T w \tag{6.1}$$

This intrinsic quality of a landing site is the same as the first term defined in expression (3.8). As mentioned in Section 5.1, the majority of the 205 possible categories are irrelevant to terrain navigation (indoor locations, i.e. showers, basements), hence only some of the weights in w are adjusted from a default value. Table 6.1 shows the (53) weighted

categories and corresponding choice as either desirable or undesirable. Favorable and unfavorable landing sites are incremented from the default weighting positively and negatively, respectively, so that higher scores correspond to better landing sites. The weighting table is then normalized such that $w \leq 1 \ \forall w$. While different weights could potentially be assigned to each type (to favor certain labels more or less, for instance), for simplicity all "good" categories are weighted the same positive amount, and all "bad" categories the same negative weight. It should be noted that some of these categories do not appear significantly in the image database for this application (see Figure 5·2), but that does not adversely affect the overall landing site scoring.

For different applications, and even different parafoil landing missions, these weightings could easily be adjusted. Moreover, optimizations could be done to figure out which combinations of categories result in the best classification of a particular map. The categories in Table 6.1 are chosen and weighted simply to demonstrate the great advantage and power afforded by having semantically meaningful categories: they can be designated in a straightforward, human-understandable, common-sense manner.

Good (+)	Bad (-)
/c/canyon 40'	/a/aqueduct 6'
/c/corn_field 53'	/a/apartment_building/outdoor 13'
/d/driveway 67'	/b/badlands 14'
/f/forest_path 78'	/b/bayou 22'
/f/forest_road 79'	/c/coast 47'
/f/field/cultivated 82'	/c/creek 58'
/f/field/wild 83'	/d/dam 63'
/g/golf_course 89'	/d/dock 65'
/h/highway 92'	/d/desert/sand 68'
/p/parking_lot 135'	/h/harbor 90'
/r/railroad_track 148'	/l/lighthouse 111'
/r/runway 160'	/m/marsh 116'
/s/shed 164'	/m/mountain 121'
/t/train_railway 185'	/m/mountain_snowy 122'
/t/track/outdoor 190'	/o/ocean 128'
/v/valley 193'	/p/pond 144'
/w/wheat_field 201'	/r/residential_neighborhood 151'
/y/yard 204'	/r/river 156'
	/r/rock_arch 157'
	/r/rope_bridge 158'
	/s/sandbar 161'
	/s/sea_cliff 163'
	/s/snowfield 173'
	/s/swamp 176'
	/s/stadium/baseball 177'
	/s/stadium/football 178'
	/s/swimming_pool/outdoor 181'
	/t/trench 187'
	/t/temple/east_asia 188'
	/t/temple/south_asia 189'
	/u/underwater/coral_reef 192'
	/v/viaduct 196'
	/v/volcano 197'
	/w/water_tower 199'
	/w/watering_hole 200'

Table 6.1: Weighted categories for landing site cost. Though some of these categories are not significantly represented in the actual dataset (Figure 5·2), this classification represents a simple, intuitive approach to choosing good and bad terrain types that can be applied more broadly.

6.2 Landing Site Classification of Satellite Imagery

To demonstrate the classification described in Section 6.1 concretely, Figure 6·1 shows the satellite map of Eloy, AZ from Figure 5·1 categorized by this approach. Lighter sections correspond to more favorable landing sites, darker to less favorable. It is noteworthy that, in general, roads and fields are ranked more favorably than desert and mountainous areas, though not all roads or fields are designated. This goes to the underlying limitations of the Places-CNN classifier, touched on in Section 5.10, which was not designed for this purpose. As these classifiers steadily improve, so to will the ability to reliably identify roadways and fields. Despite these limitations, the capability of specifying good and bad landing areas semantically in a human-readable form, and have landing classification emerge reasonably well automatically is useful.

A question one might ask is how this discretization compares to a finer one? Would smaller images result in dramatically different landing site classifications? Figure 6.2(a) demonstrates that, while a finer discretization (20x20 vs 10x10) does change the classification somewhat, the general rankings remain similar. In figure 6.2(b), each of the 4 component images in the 20x20 image are averaged to re-create a 10x10 map, which compares quite similarly to Figure 6·1. The implication here, again, is that the choice of discretization, while meaningful, does not drastically change landing site rankings, consistent with a semantic approach that incorporates understanding of the scene.

This landing site classification, in discretized satellite map space, could be used immediately by an agent seeking to land if it is able to localize using any algorithm–the one described in Chapter 5 or otherwise. In this sense, solving the localization problem immediately solves the landing problem as well. Once the agent knows where it is in this map, it's merely a matter of choosing the best landing space in that map. This echoes the discussion in Section 3.4.2. Similarly, one could follow the approach in Section 3.5 to calculate landing site optimality for all agents using the knowledge of location goodness g_y and the relative

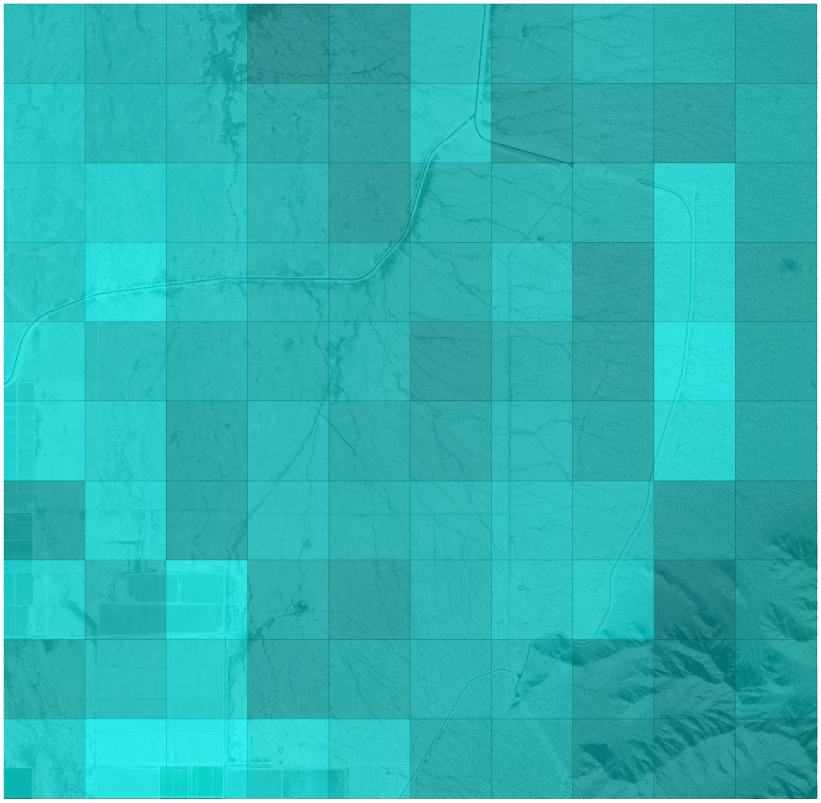
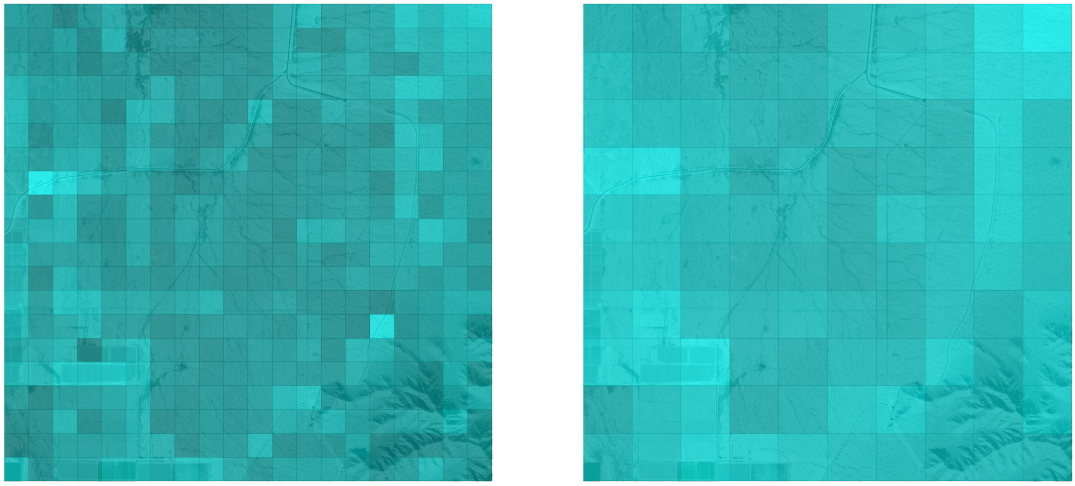


Figure 6·1: Landing location ranks calculated in satellite image space. Lighter hues correspond to more favorable locations, darker to less favorable. In general, roads and fields are ranked more favorably than desert and mountainous areas, though not all roads or fields are designated. This relates to the underlying limitations of the Places-CNN classifier, which was not designed with this purpose in mind.

distance(s) agents are away from those locations.

The more interesting scenario, however, is where an agent (or group) is unable to sufficiently localize, or alternatively, the maps available may be unreliable or unavailable entirely. In this scenario, they must rely on their ability to classify in-flight camera imagery, and rank landing sites using that alone. Section 6.3 explores these circumstances.



(a) (b)

Figure 6·2: Landing site rankings for 20x20 discretized space (left) and averaged to reconstruct at 10x10 map (right). While the rankings are not identical, they are generally similar to those in Figure 6·1, consistent with a semantic approach.

6.3 Comparing Satellite and In-Flight Camera Imagery

To examine the scenario of single and multiple agents unable to sufficiently localize, an approach reminiscent of Chapter 5 is taken. In particular, the notion of discretizing space into location tiles allows for keeping track of distinct landing locations as an agent traverses an environment. Each agent has accurate body-frame odometry, as in Section 5.6, and is thus able to keep track of the sites they have observed and classified this way as they traverse the unknown environment. This assumption is consistent with the capabilities of MSCKF, which combines IMU and tracking of visual references to provide nearly drift-free odometry [35].

6.3.1 Multi-agent Protocol

The multi-agent sharing protocol is similar in spirit as well, though with important differences. In this scenario, each agent has its own reference frame based on the path it has traversed. A common reference frame is needed for a shared knowledge representation. The

calculation of that frame would have to be based on each agent's respective localization estimate, which would then implicitly correlate successive measurements shared, and then require extensive book-keeping or other calculations to ensure consistency–all the same procedures as the approaches in Section 2.4.

Instead, similar to Section 3.6, the raw landing score measurements g_{y_h} for agent h's current image y are broadcast outward. Each agent k is able to measure the scalar distance $d_{k,h}$ away agent h's broadcast originates from (using RF ranging, for instance, just as assumed in Section 3.6.1).

The next step, however, differs from the previous approach. Since the agents have no shared frame of reference, but instead have their own, odometrically built discrete database B of locations b they have observed, classified, and scored, they can determine if any potential locations in their own database that are a distance $d_{k,h}$ away have not been visited or scored yet. Those locations in their own database would thus have no landing score, and the agent then simply populates every location a distance $d_{k,h}$ away from themselves they do not yet have a score for with the received broadcast landing score g_{y_h} from agent h.

Practically, this means that whenever an agent observes a location, it classifies and scores it itself in its own database, overwriting any previous score there. However, if the agent does not visit a location, it populates the score for that location using the first available measurement from another agent. Once a score from either another agent or its own measurement has populated a space in the database, future measurements from other agents cannot overwrite it. This is summarized in Algorithm 2.

The idea is to rely primarily on one's own measurements for landing classification, but to rapidly populate the database for far away locations it may not visit, use measurements from other agents. This is an extremely limited and rudimentary approach, by design. It

requires broadcasting literally one number (landing score) outward each time step, which is again used by other agents by measuring the scalar distance away the signal comes from to themselves. In addition to avoiding the many complications of other multi-agent approaches, it also helps accomplish the goal of landing site identification more rapidly for far away locations, even if most of the database is populated erroneously. Although the possibility exists that agents will be led astray by misleading information sharing, as they traverse their own environment and continue to sample it they will be able to avoid being led far off course in most circumstances, particularly if following a reasonable guidance strategy, such as one designed to prevent them from traveling too far away from other agents.

Algorithm 2 Multi-Agent Semantic Landing Site Classification

for all $h \in K \neq k$ **do**
  $d_{k,h} \leftarrow \mathrm{d}(x_k, x_h)$
  for all $b_i \in B$ **do**
    matches $\leftarrow \mathrm{find}(|b_i - d_{k,h}| \leq \varepsilon)$
    for all matches **do**
      if $\mathrm{match}(b_j) == 0$ **then**
        $b_j = g_{y_h}$
      end if
    end for
  end for
end for

6.3.2 Simulation Setup and Results

To compare how well agents are classifying their environment to a baseline classification of the same environment, tests are once again conducted using the camera imagery of the site in Eloy, AZ described in Section 5.2, with landing scores from the satellite imagery classified in Section 6.2 serving as a baseline.

The in-flight camera imagery is the same as described in Section 5.3. Agents observe any of a number of multiple, heading dependent camera images at at each location, corresponding to a scenario of unknown location and heading. As in Section 5.8, agents observe adjacent location images earlier in flight, corresponding to being at higher altitudes, with fewer and

		Number of Agents			
		1	2	3	4
Time Steps	10	0.8502	0.3168	0.2920	0.2865
	100	0.6798	0.2417	0.2369	0.2369
	400	0.3744	0.2308	0.2297	0.2303

Table 6.2: Landing site cost normed distance for 10x10 discretization. Theoretical minimal distance between camera and satellite imagery is 0.1973

fewer adjacent locations visible as time proceeds.

In practice, agents would be following a logical search algorithm of some kind based on exploring unknown environments, which could potentially draw upon multi-agent coverage algorithms. For the purposes of demonstration, the agents here are following random walks, just as in Section 5.9. This is certainly a sub-optimal strategy, and thus demonstrates a conservative estimate of how well this approach can work.

Landing scores estimates $\tilde{L}$ for each agent are compared to the baseline satellite scores L to get a raw measurement of how far off the camera-based classification is at each time step. These are then normalized by the norm of the baseline scores L, to get a proportional representation of how far off the camera scores are. This takes the form of expression (6.2).

$$\frac{||\tilde{L} - L||}{||L||} \tag{6.2}$$

A crucial aspect of landing site identification for parafoil systems is speed. If it takes an agent too long to determine where to land, it may run out of time and altitude to reach the best possible location. To demonstrate the dependence on both time and number of agents, Table 6.2 and Figure 6·3 show the normalized landing classification distance as a function of both. As in Section 5.9, each scenario (combination of agents and time steps) is tested with 1000 runs. For comparison, Table 6.3 shows results for the 20x20 discretization (Figure 6.2(a)).

		Number of Agents			
		1	2	3	4
Time Steps	10	0.8856	0.2908	0.2532	0.2446
	100	0.8811	0.1991	0.1931	0.1890
	400	0.6916	0.1867	0.1860	0.1838

Table 6.3: Landing site cost normed distance for 20x20 discretization. Theoretical minimal distance between camera and satellite imagery in this case is 0.1715.

6.4 Discussion

The minimum theoretical distance from camera imagery to satellite imagery in the 10x10 discretization is calculated using expression (6.2) at 0.1973. This value represents the overall distance of the Places-CNN classification of the satellite imagery and camera imagery when weighted according to expression (6.1) using Table 6.1. With sufficient time, agents are capable of approaching that value, which makes sense, given that they can directly observe more of the environment as time proceeds.

Though a single agent working alone is able to generate a landing classification that approaches the theoretical distance (Figure 6·3, column 3), it is clear that multiple agents are capable of generating closer estimates in a rapid amount of time by comparison.

The advantage of multiple agents shown for localization in Section 5.10 is once again demonstrated here in landing site classification.

Based on the results in Section 6.2, the benefits of additional agents after the first are far more muted here as compared to those in the localization context (Section 5.9). This stems from a number of factors. Most prominently, the greedy algorithm used here for landing site classification is designed to populate a database of locations as rapidly as possible. Hence even two agents will be capable of seeing and classifying the majority of the test area, and will thus substantially outperform a single agent lacking this coverage. A single agent, even with adjacent sensing, may never observe the entire environment since it must individually

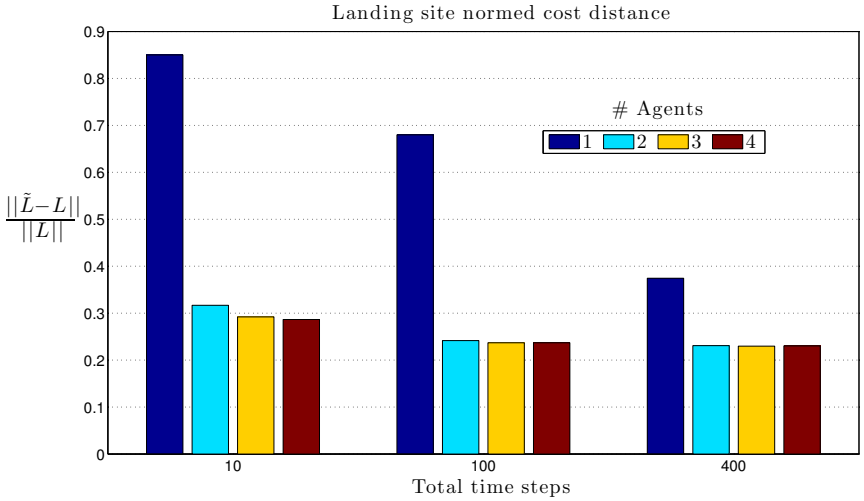


Figure 6·3: Normalized distance away of estimated landing cost function from camera images compared to true landing site cost function of environment from satellite imagery as a function of time steps and number of agents. With sufficient time steps even a single agent working alone is able to generate a landing classification that approaches truth, but it is clear that multiple agents are capable of generating closer estimates in a rapid amount of time by comparison.

navigate there to get any information about it. As a direct result, there is a large disparity between one and two agents.

In the case of more than two agents, the algorithm described in 6.3.1 is not designed to benefit from additional measurements of the same location coming from other agents, since there is no attempt to find matching coordinate frames to identify which measurements from which agent correspond to the same locations. In order to have a filtering approach similar to that pursued in Chapter 5, the agents would need to agree on a coordinate frame first, at which point they could collectively build a map that each contributes its measurements to, and those measurements would be weighted appropriately. This would be similar to work described in Section 2.4, where enough time and measurements are required to accurately identify overlapping observed landmarks to build a reference frame, at which point a larger map can be constructed and then passed through the network. All of this

is certainly possible, and future work could explore those approaches to improve upon the performance demonstrated here.

The circumstance where more than two agents could potentially perform significantly better would be in the case where agents are very far apart, and the reachability space (the cone of possible distance each can travel) is small compared to the inter-agent distance. In that circumstance, multiple agents that are a multitude of distances away from each other would be capable of populating each other's landing site databases far more rapidly than only two agents by virtue of seeing more of the test area. This is evidenced in the first column of Table 6.3, where the test environment is 20x20 rather than 10x10. With sufficient steps, however, multiple agents continue to approach much closer to the theoretical image distance than a single agent.

Chapter 7

Conclusion

The promise of visual sensing and multi-agent algorithms have been touted for years, often presented as some of the next big advances in autonomous vehicle navigation and guidance. The goal of this work has been to demonstrate how state of the art computer vision techniques, using machine learning, can make visual sensing be better, not just more complicated, than GPS for navigation. Similarly, while multi-agent localization is far from a new topic, this work presents a clear demonstration of how multiple agents can perform significantly better than single agents with very minimal communication and coordination.

A strength of the approaches described is the modularity and scalability of relatively computationally inexpensive operations to provide substantial information. It is assumed that a separate odometry estimator will be running, and a conventional EKF or factor graph back-end could similarly be proceeding at a higher frequency, receiving updates from these algorithms on a slower time scale, if appropriate. More generally, the multi-agent protocol demonstrated here is fairly minimal, by design, particularly as compared to the majority of work in this area, highlighted in Section 2.4. An interesting line of work to pursue could be seeing how much additional information from other agents could increase performance, ideally designed in an opportunistic way that would be capable of functioning both with minimal and additional information.

With regards to localization, it is quite possible that other whole-image descriptor approaches, including those mentioned in Section 3.3, whether based on other convolutional neural networks (CNNs) or otherwise, would be capable of producing similar or improved

results from those shown in Chapter 5. Indeed, applying an approach similar to that in Chapter 6 could be pursued, weighting relevant categories for a particular environment more heavily than irrelevant ones. The results in Chapter 5 thus should not be seen as the upper bound on performance, but rather as a clear demonstration of the advantages of multiple agents, and the feasibility of using a semantic classifier generally for the localization problem.

While the application discussed here has been GPS-denied guided parafoil systems, these algorithms need not be seen as limited to that specific circumstance. Indeed, even in a scenario with GPS available, a parallel semantic analysis of imagery could be invaluable in avoiding unmapped obstacles, dealing with changing conditions on the ground, and as a fail-safe if visual localization fails. Furthermore, any situation where GPS is unavailable, whether indoors or extraterrestrial, could be aided by a semantic approach as well by enabling algorithms to search for specific place signatures, rather than relying on human operators to identify them.

The ideas and concepts presented here demonstrate how the developing field of machine learning for semantic classification and understanding can be used organically in a localization and landing navigational context, together with simple multi-agent protocols. Further development of the underlying machine learning classifiers, and creating bespoke networks for analyzing satellite imagery would be immediate future work. Exploring the possibility of variable geometry map discretization, such as pursued in [60], is another logical area for further research.

Additionally, developing guidance algorithms to optimally search in both a mapped and unmapped environment with multiple agents is another direction for improving the speed and accuracy of localization and landing site identification. Agent dispersion early in flight for maximal area coverage which ensures agents remain geometrically spaced based on remaining flight time would be a logical approach to explore the trade-offs of exploration versus

exploitation (some agents could potentially be used sacrificially to explore the environment for assisting in a larger deployment as well). The active SLAM approaches described in Section 2.5 could be explored further as well, particularly since the discrete space model here is similar to much of that work.

In this scenario a top down camera view has been used, but ground-level imagery could be analyzed this way as well–both the Places-CNN classifier and general geometric constraint approach could be adapted to ground-based camera imagery compared to Google street view, for instance. Autonomous cars traveling through locations with spotty or inaccurate GPS (tunnels, overpasses) and with limited data signal to download current maps could continue operating and navigating safely based on semantic processing of exits, construction signage, and understanding unmapped hazards. Possibilities abound.

Computer vision and multi-agent robotics are among the most promising areas for further enabling autonomous navigation and exploration of the universe. We do well to remember that our imaginations, rather than the sky or GPS satellites, is the limit.

Bibliography

[1] H. Altwaijry, E. Trulls, J. Hays, P. Fua, and S. Belongie. Learning to match aerial images with deep attentive architectures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, number EPFL-CONF-217963, 2016.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[3] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, Nov. 1998.

[4] A. Bahr, M. R. Walter, and J. J. Leonard. Consistent cooperative localization. In *IEEE International Conference on Robotics and Automation, 2009. ICRA'09.*, pages 3415–3422. IEEE, 2009.

[5] D. Baronov and J. Baillieul. Decision making for rapid information acquisition in the reconnaissance of random fields. *Proceedings of the IEEE*, 100(3):776–801, 2012.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[7] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, volume 1, pages 540–545. IEEE, 2002.

[8] M. A. Brubaker, A. Geiger, and R. Urtasun. Lost! leveraging the crowd for probabilistic

visual self-localization. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3057–3064. IEEE, 2013.

[9] Y. Cao, W. Yu, W. Ren, and G. Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial informatics*, 9(1):427–438, 2013.

[10] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1412–1417. IEEE, 2013.

[11] D. Carter, L. Singh, L. Wholey, S. Rasmussen, T. Barrows, S. George, M. McConley, C. Gibson, S. Tavan, and B. Bagdonovich. Band-limited guidance and control of large parafoils. *AIAA Paper*, 2981:2009, 2009.

[12] E. Law and C. Dever. Autonomous parafoil guidance in high winds. *Journal of Guidance, Control, and Dynamics*, 38(5):963–969, 2014.

[13] E. Law and C. Dever. High wind autonomous parafoil guidance. In *23rd AIAA Aerodynamic Decelerator Systems Technology Conference*, page 2158, 2015.

[14] S. Choudhary, L. Carlone, H. I. Christensen, and F. Dellaert. Exactly sparse memory efficient slam using the multi-block alternating direction method of multipliers. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1349–1356. IEEE, 2015.

[15] L. E. Clement, V. Peretroukhin, J. Lambert, and J. Kelly. The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter. In *2015 12th Conference on Computer and Robot Vision (CRV)*, pages 23–30. IEEE, 2015.

[16] T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley & Sons, 2012.

[17] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 2042–2048. IEEE, 2007.

[18] M. Cummins and P. Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[19] A. Cunningham, V. Indelman, and F. Dellaert. Ddf-sam 2.0: Consistent distributed smoothing and mapping. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5220–5227. IEEE, 2013.

[20] A. Cunningham, M. Paluri, and F. Dellaert. Ddf-sam: Fully distributed slam using constrained factor graphs. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3025–3030. IEEE, 2010.

[21] A. Cunningham, K. M. Wurm, W. Burgard, and F. Dellaert. Fully distributed scalable smoothing and mapping with robust multi-robot data association. In *2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1093–1100. IEEE, 2012.

[22] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings*, volume 2, pages 1322–1328. IEEE, 1999.

[23] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous robots*, 8(3):325–344, 2000.

[24] M. C. Graham and J. P. How. Robust simultaneous localization and mapping via information matrix estimation. In *Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, pages 937–944. IEEE, 2014.

[25] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pages 1–8. IEEE, 2008.

[26] L. Itti, C. Koch, E. Niebur, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[28] B. J. Julian, S. Karaman, and D. Rus. On mutual information-based control of range sensing robots for mapping applications. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5156–5163. IEEE, 2013.

[29] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, page 0278364911430419, 2011.

[30] M. Kaess, S. Williams, V. Indelman, R. Roberts, J. J. Leonard, and F. Dellaert. Concurrent filtering and smoothing. In *2012 15th International Conference on Information Fusion (FUSION)*, pages 1300–1307. IEEE, 2012.

[31] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, 2001.

[32] B. Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1):191–233, 2000.

[33] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[34] M. T. Lazaro, L. M. Paz, P. Pinies, J. A. Castellanos, and G. Grisetti. Multi-robot slam using condensed measurements. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1069–1076. IEEE, 2013.

[35] M. Li and A. I. Mourikis. High-precision, consistent ekf-based visual–inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[36] D. G. Lowe. Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision, 1999.*, volume 2, pages 1150–1157. IEEE, 1999.

[37] E. Montijano, R. Aragues, and C. Sagüés. Distributed data association in robotic networks with cameras and limited communications. *IEEE Transactions on Robotics*, 29(6):1408–1423, 2013.

[38] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572. IEEE, 2007.

[39] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264–280, 2009.

[40] A. C. Murillo, G. Singh, J. Kosecka, and J. J. Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Transactions on Robotics*, 29(1):146–160, 2013.

[41] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE International Conference on Robotics and Automation, 2009. ICRA'09.*, pages 1402–1409. IEEE, 2009.

[42] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[43] M. Pfingsthorn and A. Birk. Generalized graph slam: Solving local and global ambiguities through multimodal and hyperedge constraints. *The International Journal of Robotics Research*, page 0278364915585395, 2015.

[44] B. J. Rademacher, P. Lu, A. L. Strahan, and C. J. Cerimele. In-flight trajectory planning and guidance for autonomous parafoils. *Journal of Guidance, Control, and Dynamics*, 32(6):1697–1712, 2009.

[45] A. Rosich and P. Gurfil. Coupling in-flight trajectory planning and flocking for multiple autonomous parafoils. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, page 0954410011413637, 2011.

[46] S. I. Roumeliotis and G. A. Bekey. Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00*, volume 3, pages 2985–2992. IEEE, 2000.

[47] S. I. Roumeliotis and G. A. Bekey. Collective localization: A distributed kalman filter approach to localization of groups of mobile robots. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00.*, volume 3, pages 2958–2965. IEEE, 2000.

[48] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[49] A. Rubio, M. Villamizar, L. Ferraz, A. Penate-Sanchez, A. Ramisa, E. Simo-Serra, A. Sanfeliu, and F. Moreno-Noguer. Efficient monocular pose estimation for complex 3d models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1397–1402. IEEE, 2015.

[50] M. Schwager, P. Dames, D. Rus, and V. Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. In *Proceedings of the International Symposium on Robotics Research (ISRR 11)*, 2011.

[51] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[52] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(4):623–656, Oct 1948.

[53] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[54] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim. Integrated position estimation using aerial image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):1–18, 2002.

[55] G. M. Siouris. *Missile guidance and control systems*. Springer, 2004.

[56] T. J. Steiner and T. M. Brady. Vision-based navigation and hazard detection for terrestrial rocket approach and landing. In *2014 IEEE Aerospace Conference*, pages 1–8. IEEE, 2014.

[57] N. Sünderhauf, F. Dayoub, S. Shirazi, B. Upcroft, and M. Milford. On the performance of convnet features for place recognition. *arXiv preprint arXiv:1501.04158*, 2015.

[58] N. Trawny, A. I. Mourikis, S. I. Roumeliotis, A. E. Johnson, and J. F. Montgomery. Vision-aided inertial navigation for pin-point landing using observations of mapped landmarks. *Journal of Field Robotics*, 24(5):357–378, 2007.

[59] M. Ward, C. Montalvo, and M. Costello. Performance characteristics of an autonomous airdrop system in realistic wind environments. In *AIAA Atmospheric Flight Mechanics Conference*, page 7510, 2010.

[60] T. Weyand, I. Kostrikov, and J. Philbin. Planet-photo geolocation with convolutional neural networks. *arXiv preprint arXiv:1602.05314*, 2016.

[61] Z. Yan, N. Jouandeau, and A. A. Cherif. A survey and analysis of multi-robot coordination. *International Journal of Advanced Robotic Systems*, 10, 2013.

[62] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.

Curriculum Vitae